

# **Enhanced Cal Poly SuPER System Simulink Model**

A Thesis  
Presented to the Faculty of  
California Polytechnic State University,  
San Luis Obispo

In Partial Fulfillment  
of the Requirements for the Degree of  
Master of Science in Electrical Engineering

by  
Matthew O. McFarland  
June 2010

© 2010  
Matthew O. McFarland  
ALL RIGHTS RESERVED

## **COMMITTEE MEMBERSHIP**

Title: Enhanced Cal Poly SuPER System Simulink Model

Author: Matthew O. McFarland

Date Submitted: June 11, 2010

COMMITTEE CHAIR: James Harris, Professor

COMMITTEE MEMBER Ali Shaban, Professor

COMMITTEE MEMBER: Dale Dolan, Assistant Professor

# **ABSTRACT**

## **Enhanced Cal Poly SuPER System Simulink Model**

Matthew McFarland

The Cal Poly Sustainable Power for Electrical Resources (SuPER) project is a solar power DC distribution system designed to autonomously manage and supply the energy needs of a single family off-the-grid home. The following thesis describes the improvement and re-design of a MATLAB Simulink model for the Cal Poly SuPER system. This model includes a photovoltaic (PV) array, a lead-acid gel battery with temperature effects, a wind turbine model, a re-designed DC-DC converter, a DC microgrid, and multiple loads. This thesis will also include several control algorithms such as a temperature controlled thermoelectric (T.E.) cooler, intelligent load switching, and an intelligent power source selector. Furthermore, a seven day simulation and evaluation of the results are presented. This simulation is an important tool for further system development, re-design, and long term system performance prediction.

## **ACKNOWLEDGMENTS**

I would like to acknowledge Dr. Ali Shaban and Dr. James Harris for their guidance and support throughout my work on the SuPER project. I would also like to thank Professor Dale Dolan for reviewing this thesis. I wanted to thank my family and my fiancé Chelsey Brown for putting up with me and the many long nights, weekends, and holidays I spent working on this thesis.

Matthew O. McFarland  
June 2010

# TABLE OF CONTENTS

LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
Chapter 1: Introduction .....	1
1.0 SUPER PROJECT INTRODUCTION .....	1
1.1 THE OVERALL MODEL .....	1
1.2 THESIS OBJECTIVES .....	5
1.3 DOCUMENT OVERVIEW .....	5
Chapter 2: Background .....	7
2.0 CHAPTER OVERVIEW .....	7
2.1 BATTERY MODELING .....	7
2.1.1 Basic Lead-Acid Battery Chemistry .....	7
2.1.2 Resistive Companion Battery Modeling .....	10
2.1.3 Battery Modeling Energy Aware System Design .....	12
2.1.4 Battery Management Based Upon Non-Linear Dynamics Modeling .....	14
2.2 DC MICROGRID .....	15
2.3 WIND TURBINE .....	17
2.4 THERMOELECTRIC COOLER .....	20
Chapter 3: Enhanced SuPER MATLAB Simulink Model .....	23
3.0 CHAPTER OVERVIEW .....	23
3.1 PLANS FOR ENHANCED SIMULATION .....	23
3.2 SIMULINK MODELS .....	26
3.2.1 Battery Model .....	26
3.2.2 DC Microgrid Modeling .....	32
3.2.3 Wind Turbine Design and Modeling .....	47
3.2.4 New PV DC-DC Converter Model .....	55
3.2.5 PV Control Algorithm Modification .....	59
3.2.6 Spartan 3E Control Board (FPGA) .....	59
3.2.7 Satellite Modem and Wireless Router .....	60
3.2.8 New DC Motor Model .....	62
3.2.9 New Thermoelectric Model .....	65
3.2.10 Switch Model and Intelligent Load Control .....	69
Chapter 4: Simulation .....	73
4.0 CHAPTER OVERVIEW .....	73
4.1 SETTING UP THE SIMULATION ENVIRONMENT .....	73
4.2 SAMPLE TIMES AND DATA RECORDING .....	74
4.3 REQUIREMENTS FOR ONE WEEK SCENARIOS .....	76
4.3.1 Possible Load Schedules .....	76
4.3.2 Input Data .....	78
4.3.3 Assumptions .....	80
4.3.4 Load Scenario Number .....	80

4.4	SCENARIOS .....	81
4.4.1	PV Standalone .....	81
4.4.2	PV and Microgrid.....	81
4.4.3	Wind Turbine Standalone .....	81
4.4.4	Wind Turbine and Microgrid .....	81
4.4.5	PV and Wind Turbine Standalone.....	81
4.4.6	PV, Wind Turbine, and Microgrid .....	82
4.5	RESULTS .....	82
4.5.1	PV Standalone .....	82
4.5.2	PV and Microgrid.....	83
4.5.3	Wind Turbine Standalone .....	83
4.5.4	Wind Turbine and Microgrid .....	84
4.5.5	PV and Wind Turbine Standalone.....	84
4.5.6	PV, Wind Turbine, and Microgrid .....	85
4.5.7	Summary of Results .....	85
4.6	EVALUATION: .....	87
4.6.1	PV Standalone .....	87
4.6.2	PV and Microgrid.....	87
4.6.3	Wind Turbine Standalone .....	88
4.6.4	Wind Turbine and Microgrid .....	88
4.6.5	PV and Wind Turbine Standalone.....	89
4.6.6	PV, Wind Turbine, and Microgrid. ....	89
4.6.7	Data Summary.....	90
Chapter 5:	Conclusion.....	92
5.1	ACHIEVEMENTS .....	92
5.2	FUTURE RECOMMENDATIONS .....	93
5.2.1	T.E. Cooling for the Battery to Extend Battery Life.....	93
5.2.2	Implement Parallel Processing for Speed Improvements in the Simulation. ....	94
5.2.3	Frame Based Processing to Increase Simulation Speed.....	94
5.2.4	Battery Discharge Characteristic Tracking .....	95
5.2.5	Increase DC-DC Converter Accuracy.....	95
5.2.6	Implement in Software the Intelligent Load Control .....	95
5.2.7	Create a Better Power Source Selection Algorithm .....	96
Bibliography	.....	97
APPENDIX A:	C-MEX S-FUNCTION CODE.....	101
A.1	Battery Model S-Function Improved Code .....	101
A.2	DC Microgrid S-Function Code .....	102
A.3	Hut Scenario Load Control Block S-Function Code.....	104
A.4	Wind Turbine S-Function Code .....	105
A.5	PV DC-DC Converter Control S-Function Block Code .....	106
A.6	PV Charge Control Block S-Function Code .....	107
A.7	Motor Model Control Block S-Function Block Code .....	108
A.8	T.E. Cooler Thermal Model S-Function Code .....	109
A.9	T.E. Cooler Electric Model S-Function Code .....	110
A.10	Scenario Load Control S-Function Block Code.....	111

APPENDIX B: WIND DATA PROCESSING MACROS .....	113
B.1 Wind Data Raw Processing Macro (Written in VBA) .....	113
B.2 Wind Data All Seven Days Algorithm (Written in VBA) .....	114
APPENDIX C: MATLAB/SIMULINK DATA MANIPULATION COMMANDS AND FUNCTIONS .....	120
C.1 Using From File Block .....	120
APPENDIX D: SPEED TESTING .....	125
D.1 DC-DC Converter Attempted New Model .....	125
D.2 DC-DC Converter Final Model Version .....	126
APPENDIX E: COMPILER SETUP .....	127
APPENDIX F: README FILE FOR DVD .....	130



## LIST OF TABLES

Table 2.1: Battery Model Table [5] .....	13
Table 3.1: Efficiency Table Example .....	34
Table 3.2: Air X Wind Turbine Voltage Regulation Table .....	49
Table 3.3: Air X Wind Turbine Wire Gauge Chart [28].....	51
Table 3.4: Thermal Model Table .....	67
Table 4.1: Data Recording Items .....	74
Table 4.2: Simulink Limits Table .....	76

## LIST OF FIGURES

Figure 1.1: Simulation Top Level Diagram [1] .....	2
Figure 1.2: Previous Simulink Simulation [1] .....	4
Figure 2.1: Battery SOC Estimate [4].....	15
Figure 2.2: Dispersed Control System Diagram [6] .....	16
Figure 2.3: Wind Turbine Control Setup [9] .....	17
Figure 2.4: Wind and Fuel Cell Microgrid [10].....	19
Figure 2.5: Coleman Thermoelectric Cooler [11].....	20
Figure 2.6: Thermoelectric Cooler Circuit Diagram.....	21
Figure 2.7: T.E. Effect Diagram [14].....	22
Figure 2.8: Typical T.E. Cooler Setup [14] .....	22
Figure 3.1: Enhanced Simulink Simulation Model .....	25
Figure 3.2: Basic Battery Model.....	27
Figure 3.3: Temperature Relationship Graphs [17] .....	30
Figure 3.4: Battery Discharge Data [1].....	31
Figure 3.5: Battery Self Discharge Modeling [1] .....	31
Figure 3.6: Battery Charge Data [1] .....	32
Figure 3.7: Photo of an African Village [19].....	33
Figure 3.8: Efficiency Example .....	34
Figure 3.9: Microgrid Transmission Diagram .....	36
Figure 3.10: Typical Cell Phone Charger .....	37
Figure 3.11: Power Consumption View of a Cell Phone Charger [20] .....	37
Figure 3.12: Sangean Radio [21] .....	38
Figure 3.13: LED Light Bulb [22] .....	39
Figure 3.14: Lumped Load Diagram .....	40
Figure 3.15: Hut Power Consumption Graph .....	42
Figure 3.16: Hut Loads Block Diagram.....	43
Figure 3.17: Buck ZVS Circuit [25] .....	44
Figure 3.18: ZVS Waveforms [25] .....	44
Figure 3.19: DC Microgrid Block Diagram.....	45
Figure 3.20: DC Microgrid Detail .....	46
Figure 3.21: Air X Wind Turbine Dimensions [28] .....	48
Figure 3.22: Air X Wind Turbine Picture [28] .....	48
Figure 3.23: Air X Wind Turbine Power Output Graph [27] .....	50
Figure 3.24: Air X Wind Turbine Power Curve Fitting .....	51
Figure 3.25: Wind Turbine Simulink Block .....	52
Figure 3.26: Raw Wind Data File .....	53
Figure 3.27: Processed Wind Data.....	54
Figure 3.28: Old DC-DC Converter Model [1] .....	55
Figure 3.29: Circuit Diagram of New DC-DC Converter [32].....	56
Figure 3.30: Re-Created DC-DC Converter in Simulink.....	57
Figure 3.31: DC-DC Converter Block Diagram.....	57
Figure 3.32: DC-DC Converter Efficiency Data [32].....	58
Figure 3.33: Xilinx Power Estimator Worksheet.....	60
Figure 3.34: Direct Way Satellite Modem [34] .....	61

Figure 3.35: Direct Way Satellite [34].....	61
Figure 3.36: Cisco Wireless Router [35] .....	61
Figure 3.37: Simulink Model of a DC Motor [36].....	62
Figure 3.38: Previous DC Motor Model [1] .....	62
Figure 3.39: DC Motor In-rush Current.....	63
Figure 3.40: Simulink Implementation of DC Motor .....	64
Figure 3.41: DC Motor Model Transient Output Current.....	65
Figure 3.42: DC Motor Model Transient Output Current Zoom In.....	65
Figure 3.43: Electric Model of T.E. Cooler .....	66
Figure 3.44: T.E. Cooler Thermal Model .....	67
Figure 3.45: T.E. Smart Controller Block Diagram.....	68
Figure 3.46: T.E. Temperature Control Results.....	69
Figure 3.47: Switch and Loads Block Diagram.....	70
Figure 3.48: Switch and Loads Detail.....	71
Figure 3.49: Load Logic and Gate Control Block Diagram .....	72
Figure 4.1: Primary Load Schedule Scenarios 1 and 2 .....	77
Figure 4.2: Primary Load Schedule Scenarios 3 and 4.....	77
Figure 4.3: Primary Load Schedule Scenarios 5 and 6.....	77
Figure 4.4: Microgrid Load Schedule .....	78
Figure 4.5: PV Insolation Data Golden, CO [7] .....	78
Figure 4.6: Normalized Wind Speed Input Data San Luis Obispo, CA [30].....	79
Figure 4.7: Ambient Temperature Data Golden, CO [7] .....	79
Figure 4.8: PV Stand Alone SOC Results .....	82
Figure 4.9: PV Microgrid SOC Results .....	83
Figure 4.10: Wind Turbine Standalone SOC Results .....	83
Figure 4.11: Wind Turbine and Microgrid SOC Results.....	84
Figure 4.12: PV and Wind Turbine SOC Results.....	84
Figure 4.13: PV, Wind, and Microgrid SOC results.....	85
Figure 4.14: SOC Summary Chart.....	85
Figure 4.15: Loads Serviced Chart Microgrid Off.....	86
Figure 4.16: Loads Serviced Chart Microgrid On .....	86

# **Chapter 1: Introduction**

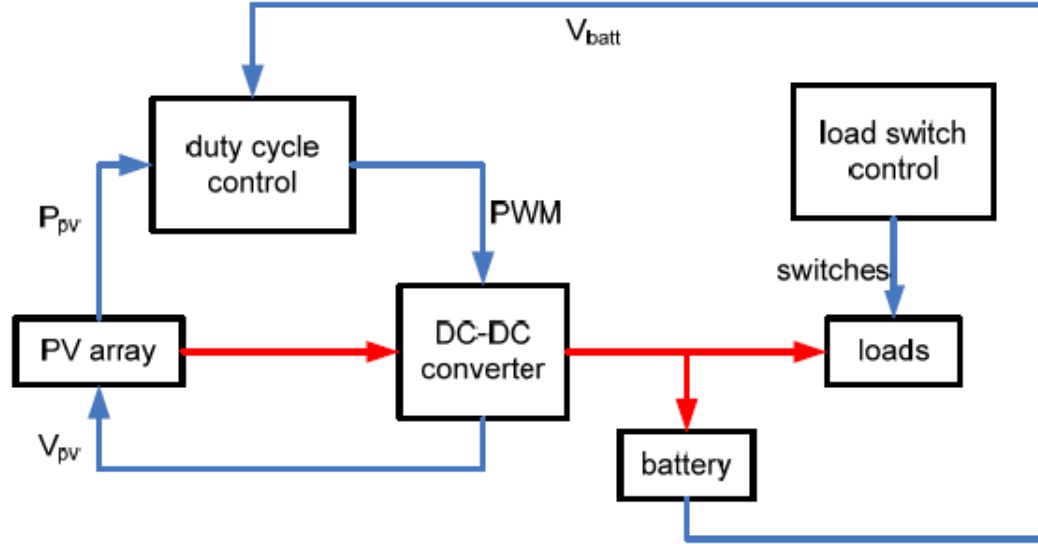
## **1.0 SuPER Project Introduction**

The Cal Poly Sustainable Power for Electrical Resources (SuPER) project can be viewed as an autonomous power generation and distribution system. The power is generated through a 150W photovoltaic panel, and the voltage is then stepped down to a 12V bus voltage.

Next, the various loads are supplied with power, and during times of no sunlight the battery provides power to the required loads. The advantage of the Cal Poly SuPER system is that every component can be independently controlled and monitored, which means that the power use and system life can be optimized. Through measurement, analysis, simulation, and evaluation the system can be made more robust and efficient.

## **1.1 The Overall Model**

Prior to beginning simulation work on the Cal Poly SuPER system a Simulink Model had been written by Tyler Sheffield in cooperation with many of the previous senior projects and Master's thesis projects[1]. The model incorporated many of the components and framework that the system was made up of at the time. A basic top level diagram of what was implemented is shown in Figure 1.1.



**Figure 1.1: Simulation Top Level Diagram [1]**

The previous model included a simple model of a DC-DC converter, a battery model, a model of the loads (DC motor, LED lighting, thermoelectric (T.E.) cooler, TV, and Laptop) and the photovoltaic model.

In order to improve the simulation the existing Simulink model was evaluated. The previous version was written in MATLAB 2007 and at the time the only available version was MATLAB 2008b. It was determined that the existing simulation was not compatible with MATLAB 2008b.

Furthermore, after corresponding with Sheffield [1] it was discovered that the version of the simulation that was being used was incomplete. Upon obtaining the complete version it was discovered that there were more errors. By this point the first quarter was almost over. The decision was made to start a new simulation using the same structure, framework, and mathematical models as the previous model and then make improvements to this model.

To develop the new Simulink Model an analysis was done on the non-functioning simulation [1]. Using the previous Simulink model (Figure 1.2) as a general template every block was rebuilt.



## **1.2 Thesis Objectives**

The primary objective of this thesis is to improve the speed and accuracy of the simulation model in addition to adding more capabilities to the model. The second objective is to run a seven day simulation. The objectives will be met by making multiple changes which involve creation of a new battery model that includes ambient temperature effects, creation of a wind turbine model based upon a commercially available design, model implementation of a DC-microgrid, creation of a new charge control algorithm for the PV array, addition of several new load models, addition of an intelligent load switching algorithm model, addition a temperature controlled T.E. cooler model, addition of a smart power source selector, and the simulation and evaluation of seven days worth of real input data.

## **1.3 Document Overview**

Chapter two in large part is dedicated to the research done regarding the modeling of lead-acid batteries. Chapter two also touched on the research done on wind turbines, microgrids, and some background information on thermoelectric cooling.

Chapter three includes the bulk of the work in this thesis. In this chapter the previous simulation is discussed along with the plans to design a new simulation. All the justifications, designs, and implementations of the various Simulink models are discussed in detail in this chapter.

Chapter four covers a first time user's guide in using the simulation, assumptions in running the simulation for a seven day scenario, results for the seven day simulation, and an evaluation of the results.



The final chapter brings everything together by summarizing what was accomplished along with future recommendations. The recommendations build upon the knowledge gained from working on the SuPER project simulation. The appendices include information to give extra details about the simulation; however, the information items are considered supplementary.

## Chapter 2: Background

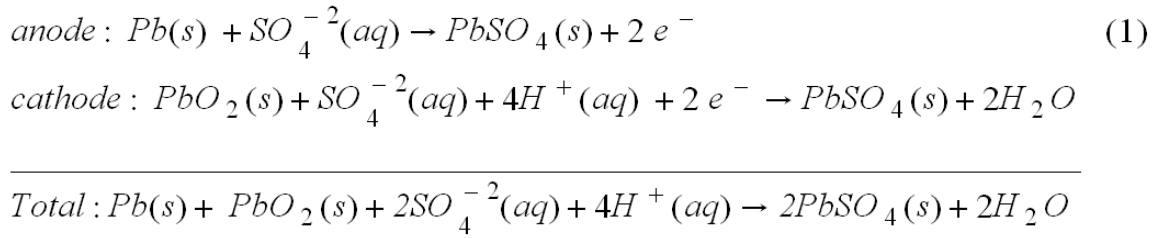
### 2.0 Chapter Overview

This chapter will cover the basics of lead-acid battery chemistry, along with previous research in battery modeling. Also, this chapter will cover studies dealing with microgrids, wind turbine control, and the basic theory and operation of a thermoelectric cooler.

### 2.1 Battery Modeling

#### 2.1.1 Basic Lead-Acid Battery Chemistry

Lead-acid battery chemistry is surprisingly complex; however the equations will be broken up to simplify the things. All electro-chemistry reactions are broken up into cell potential reactions. In the case of a lead-acid battery discharging, the following reactions are shown [2]:



As shown in (1), the electrolyte or charge carrier is the sulfuric acid. The anode is a lead grill filled with spongy lead, while the cathode is a lead grill coated with lead oxide. A typical 12V lead-acid battery is made up of 6 cells arranged in series (each cell potential being about 2V) [2].

At the anode during discharge the lead and the sulfuric acid ions combine to form lead sulfate and release electrons in the reaction. In the case of the cathode the lead oxide and the sulfuric acid ions combine to form lead sulfate and water. As the battery is discharged more lead sulfate and water is formed as the battery becomes depleted, therefore the sulfuric acid concentration will drop. To reverse the reaction (charging) the reactions are the same except the arrows are reversed. For any point in the charge or discharge cycle of the battery, the state of charge of the battery is said to be proportional to the concentration of the acid in the battery at any given time. Therefore, if discharging, the acid concentration will fall and if charging the acid concentration will rise.

To determine the cell potential or voltage of any given cell the Nernst equation (2) is used [2].

Nernst Equation : (2)

$$\Delta E = \Delta E^\circ - \left( \frac{RT}{nF} \right) \ln Q \quad (V)$$

$\Delta E^\circ$  – standard cell voltage (V)

$T$  – temperature in (K)

$F$  – faradays constant (96, 485 C mol<sup>-1</sup>)

$R$  – 8.315 (JK<sup>-1</sup> mol<sup>-1</sup>)

$n$  – moles of electrons (mol)

$Q$  – Reaction quotient (mol)

The Nernst equation (2) is very useful in calculating how the battery will react under a variety of conditions. The battery voltage can be affected by temperature and the reaction quotient (Q term). The value of the Q term is not available because the battery

in use is a sealed lead-acid battery. Therefore, the battery chemistry is an unknown variable and an equivalent term will have to be used instead.

Since the battery is not ideal, it is always considered to have some sort of internal resistance. Now from Hans Bode [3] the resistance equation (3) is presented.

$$R_i^* = R_i + R_p = R_i + \frac{\eta}{I} \quad (3)$$

$R_i$  – resistance of conductors ( $\Omega$ )

$R_p$  – polarization resistance ( $\Omega$ )

$\eta$  – overpotential (V)

$I$  – current flow (A)

As stated by Hans Bode [3], the magnitude of the polarization resistance is roughly equal to the resistance from all of the conductors. Now, when the battery is sitting unconnected to any load, there is a property called self discharge. According to Hans Bode, self discharge can be affected by a variety of factors, including temperature, and the initial concentration of the acid inside the battery [3]. The equation presented in (4) from Hans Bode will not include temperature.

$$\text{Self discharge equation :} \quad (4)$$

$$I_s = cK_o \quad (A)$$

$K_o$  – initial starting capacity (Ahr)

$c$  – self discharge coefficient ( $\text{hr}^{-1}$ )

Since the above equation is a constant current drain it can be lumped in terms of an equivalent resistance as presented in equation (5).

$$R_{scd} = \frac{V_{batt}}{I_s} \quad (5)$$

$V_{batt}$  is the instantaneous battery voltage at any given time. The cell capacity, or rather how many AHr a cell can store, is dependent on a variety of factors including: construction, maintenance, battery age, and temperature. The best mathematical description of the capacity is called Peukert's Law [2].

$$I^n t = C \quad (6)$$

*Peukert's Equation*  
 *$n$  – constant from experimental data*  
 *$t$  – time(hr)*  
 *$C$  – capacity (Ahr)*  
 *$I$  – discharge current (A)*

It is important to note that Peukert's equation (6) does not include any temperature dependence other than the  $n$  term. Since temperature dependence is not included in the equation it may affect a model's accuracy for varying temperatures.

### 2.1.2 Resistive Companion Battery Modeling

One of the papers about battery modeling took broke up the battery modeling process by viewing the battery as a set of resistors and capacitors [4]. The original electro-chemical model is utilized to reflect the principle components. Another item to

note was that this model was written using a piece of software called VTB or virtual test bed.

The authors start with the most basic equation (two terminal model):

$$v = E_{eq} - iR_{int} \quad (7)$$

Next, they assert from the Nernst equation (2) that:

$$E_{eq} = f(SOC) \quad (8)$$

$$E_{eq} = E_o + \frac{RT}{nF} \ln \left( \frac{SOC}{1 - SOC} \right)$$

Where,

$$\begin{aligned} E_{eq} &- \text{Equilibrium voltage} && (V) \\ SOC &- \text{state of charge of battery} && (\%) \\ v &- \text{battery voltage} && (V) \\ i &- \text{battery current} && (A) \\ R_{int} &- \text{internal resistance of battery} && (\Omega) \\ E_o &- \text{Open circuit battery voltage} && (V) \end{aligned}$$

Next, from Peukert's equation (6) they state that:

$$\frac{dSOC}{dt} = \frac{i}{\left( 3600 \frac{\text{sec}}{\text{hr}} \right) C} \quad \left( \frac{\%}{\text{sec}} \right) \quad (9)$$

$$\begin{aligned} i &- \text{current out the the battery} && (A) \\ C &- \text{capacity} && (A - \text{hr}) \end{aligned}$$

Now, from these building blocks the author's elaborate more on the equations regarding, gassing, sulfate build up, and interface reactions. All of the extra effects were included in the VTB software environment. The model was complex and accurate; however as noted by the authors, the simulation speed would be limited in a different simulation environment.

### **2.1.3 Battery Modeling Energy Aware System Design**

Another paper dealing with energy aware system design actually encompassed a good overall study of the many different model approaches [5]. There are three major categories of model types discussed in the paper: empirical models, abstract models, and mixed models. Empirical models dealt with electro-chemical equations with the constants fitted to experimental data. Abstract models, like the resistive companion model, attempted to model the battery as electric circuits with empirical equations. Finally, the mixed models use a combination of abstract and empirical methods. The paper presented many methods to model batteries and the results are shown in Table 2.1.

**Table 2.1: Battery Model Table [5]**

Table 1. Battery models and applications.							
Model	Temperature effect	Capacity fading	Accuracy	Computational complexity	Configuration effort	Analytical insight	Applications
<b>Physical</b>							
Lithium-polymer-insertion cell (Doyle et al.)	Yes	Yes; support for Arrhenius temperature dependence and cycle aging added by Rong and Pedram	Very high	High	Very high (> 50 parameters)	Low	
<b>Empirical</b>							
Peukert's law	Yes; needs recalibration for each temperature	No	Medium (14% average error for constant load, 8% average error for interrupted and variable loads)	Low	Low (2 parameters)	Low	
Battery efficiency (Pedram and Wu)	Yes; needs recalibration for each temperature	No	Medium	Low	Low (2 parameters)	Low	Design of interleaved dual-battery power supply; load splitting for maximum lifetime of multibattery systems
Weibull fit (Syracuse and Clark)	Yes	No	Medium	Low	Low (3 parameters)	Low	
<b>Abstract</b>							
Electrical-circuit (Gold)	Yes	Yes	Medium (12% error predicting cell voltage and thermal characteristics, 5% error predicting cycle aging)	Medium	Medium (> 15 parameters)	Medium	
Electrical-circuit (Bergveld et al.)	Yes	No	Medium	Medium	High (> 30 parameters)	Medium	Thermostatic charge method; high charging efficiency
Discrete-time (Benini et al.)	Yes	No	Medium (1% compared to Hspice continuous-time model)	Medium	Medium (>15 parameters)	Medium	Dynamic Power Management; multibattery discharge
Stochastic (Chiasserini and Rao)	No	No	High (1%)	Low	Low (2 parameters)	Medium (stochastic model of load pattern assumed)	Shaping load pattern to exploit charge recovery
<b>Mixed</b>							
Analytical high-level (Rakhmatov et al.)	No	No	High (5%)	Medium	Low (2 parameters)	High	Task scheduling by sequencing and V/f scaling; analysis of discharge methods for multibattery systems
Analytical high-level (Rong and Pedram)	Yes	Yes	High (3.5%)	Medium	Medium (> 15 parameters)	High	

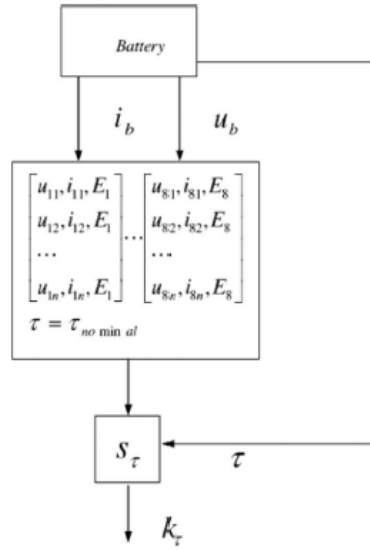


Table 2.1 illustrates clearly the many methods that can be taken in modeling and the expected level of accuracy for each method. The table became a useful tool in deciding the modeling technique to be used in the new Simulink Model.

#### **2.1.4 Battery Management Based Upon Non-Linear Dynamics Modeling.**

The modeling approach taken in this paper can be classified as an empirical battery model [4]. The derivation starts with the basic equations, which include internal resistance, Peukert's capacity equation and the Nernst equation. However, in this model an attempt is made to incorporate the effects from temperature. Furthermore, this model uses experimental data and polynomials to determine the shape of internal resistance and polarization EMF curves.

Using the model [4], battery voltage was predicted to within 1% accuracy of the actual data. More interesting to note, an algorithm is suggested to yield an accurate estimation of the state of charge of the battery. The algorithm focuses on lookup tables to produce multiple constants. The constants are multiplied by a polynomial to change the shape of the resistance and EMF curves. The method suggested is fast, efficient and as mentioned, can easily be implemented on a microcontroller [4]. The primary issue is gathering enough data to populate the lookup tables. In order for the particular method to be used in the SuPER system, much more battery run time data would need to be gathered to populate the lookup tables. Listed in Figure 2.1 is the suggested SOC algorithm.

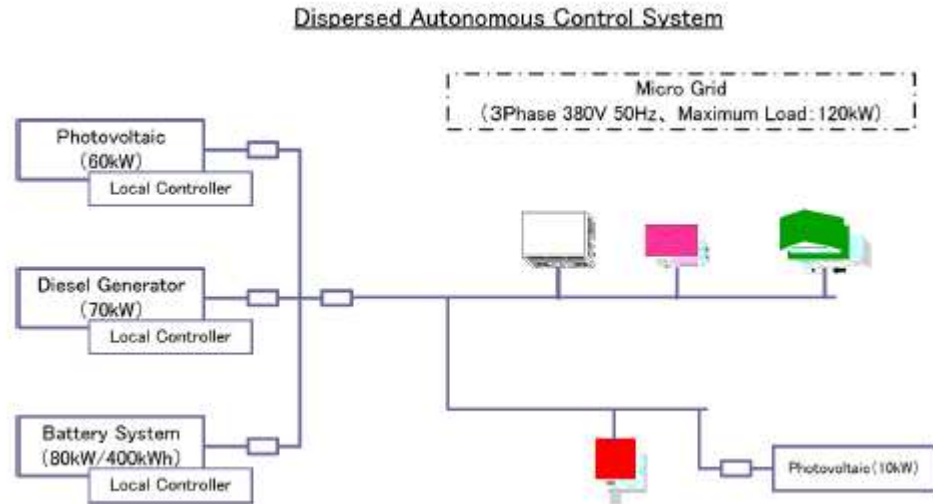


**Figure 2.1: Battery SOC Estimate [4]**

The algorithm shown above takes in the battery current, voltage, and a particular tau. Next, the data is translated into an array which uses a lookup table to yield the values to predict the battery behavior. The SOC estimation methods presented in the paper [4] would be useful in re-writing the battery SOC estimation code that currently exists on the laptop.

## 2.2 DC Microgrid

Research material specifically on DC microgrids was rare. However, information on AC microgrids was quite plentiful. A particular study done in China deals with a diesel generator, a battery bank and a photovoltaic array in a microgrid structure [6].



**Figure 2.2: Dispersed Control System Diagram [6]**

The study showed how autonomous control algorithms were created that did not use centralized control to accomplish economical load dispatching. Applying this to the system model case, the study shows how simple algorithms can be used to efficiently manage multiple loads depending on the power generation and intelligent use of a battery system. For example, when the photovoltaic array is generating enough excess power, the diesel generator will be turned off [6]. This rule minimizes the amount of fuel used by the diesel generator and therefore saves money. Implementation of power management techniques will ensure system efficiency and thus become highly applicable to the SuPER system.

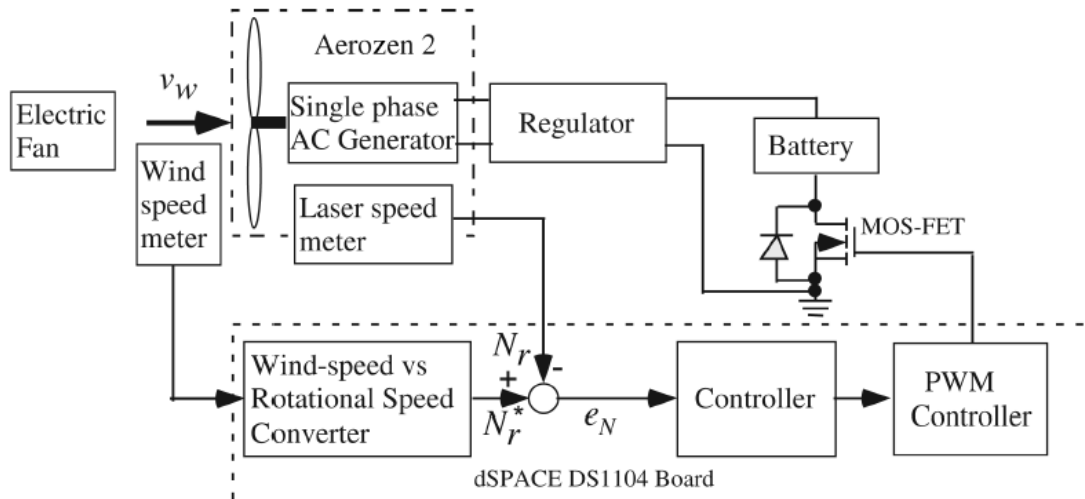
The National Renewable Energy Resource Laboratory (NREL) has a wealth of knowledge and information regarding solar and wind systems and models. In fact all of the insolation and temperature data for this project was obtained from NREL's website [7]. Just to give an idea of how much data is available; NREL has solar insolation data from Golden Colorado that is available from October 19, 2001 to September 30, 2008 (updated every minute). Furthermore, the site has a program called HOMER that is

specifically meant to evaluate microgrids in a simulation environment. The HOMER program is completely free and open source.

The nature of the Cal Poly SuPER system naturally led to study of the layout of the international space station (ISS). Both systems are DC microgrids with photovoltaic arrays, batteries, and a variety of loads to service. It was discovered that the NASA website had a manual which detailed the layout and electrical design of the international space station [8]. After reading about the use of multiple high voltage DC-DC converters being used on the ISS, it further reinforced the idea that creating a high voltage DC microgrid for the SuPER system was entirely feasible.

## 2.3 Wind Turbine

The most efficient wind turbine devices incorporate intelligent control systems to maximize the power output. In a Japanese paper a control system is designed and tested that essentially regulates the rotor speed to its most optimum velocity so that in all cases the wind generator will output maximum power [9]. The test setup is shown in Figure 2.3.



**Figure 2.3: Wind Turbine Control Setup [9]**

The optimization equations start with equation (10) [9] in which mechanical power of the wind turbine in watts is:

$$P = \eta C_p \left( \frac{1}{2} \rho A V_w^3 \right) \quad (w) \quad (10)$$

$\eta$  – Efficiency of direct drive turbine

$C_p$  – Power Coefficient

$\rho$  – Density of air ( $\text{Kg}/\text{m}^3$ )

$A$  – Blade swept area ( $\text{m}^2$ )

$V_w$  – Velocity of the wind ( $\text{m}/\text{s}$ )

Next, the authors assert that when the pitch angle is fixed the power coefficient will be based upon the tip speed ratio [9], which is:

$$\lambda = \frac{r\omega}{V_w} \quad (11)$$

$\lambda$  – tip speed ratio (rad)

$r$  – wind mill radius (m)

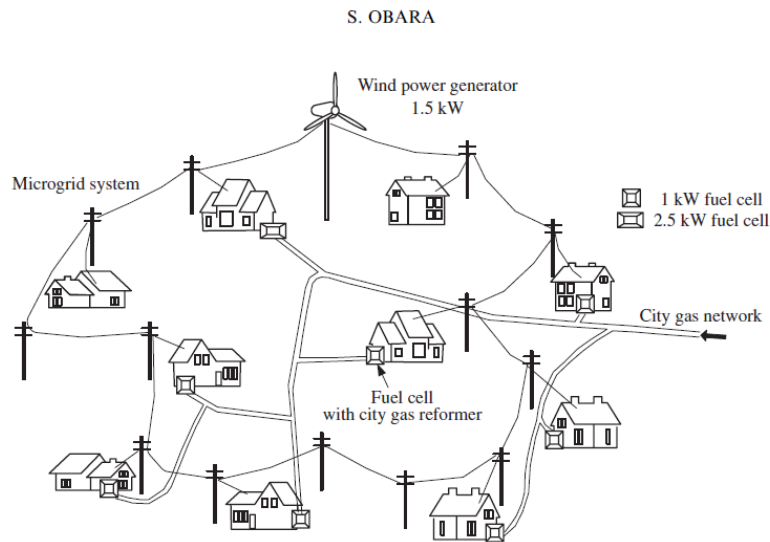
$\omega$  – angular speed ( $\text{rad}/\text{s}$ )

$V_w$  – wind velocity ( $\text{m}/\text{s}$ )

Finally, the authors state that to find the maximum power output they let  $\eta$  equal 100% and they make sure that the speed ratio matches the optimum speed ratio; therefore, in this way the maximum power output should be obtained [9]. The rotor speed is controlled by changing the loading on the generator via a battery and a pulse width modulation (PWM) signal. For example, if the rotor speed is faster than the optimum

rotor speed then the PWM signal would be changed to allow more current out of the battery and therefore an increased load would be seen by the generator. The increased load would cause the rotor speed to slow back towards the optimum rotor speed.

Another journal article dealing with wind turbines essentially evaluates the performance of a fuel cell along with a wind turbine [10]. In the case studied the microgrid is AC not DC.

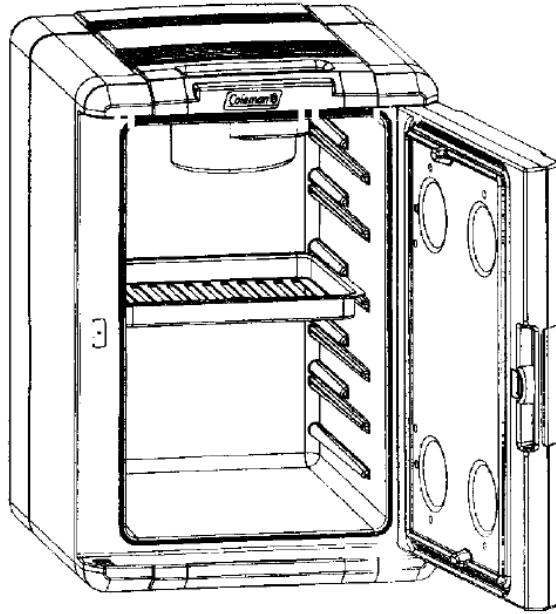


**Figure 2.4: Wind and Fuel Cell Microgrid [10]**

The paper tackles the primary problem of most renewable energy resources, which is trying to create a stable power supply from a variable power source. The only way, of course, is to have some sort of storage capacity. Just like a capacitor stabilizes output voltages, a battery or fuel cell system must stabilize power generation. In the case studied it is shown that a fuel cell performs much better than a equivalent battery system because of the larger energy storage capacity [10].

## 2.4 Thermoelectric Cooler

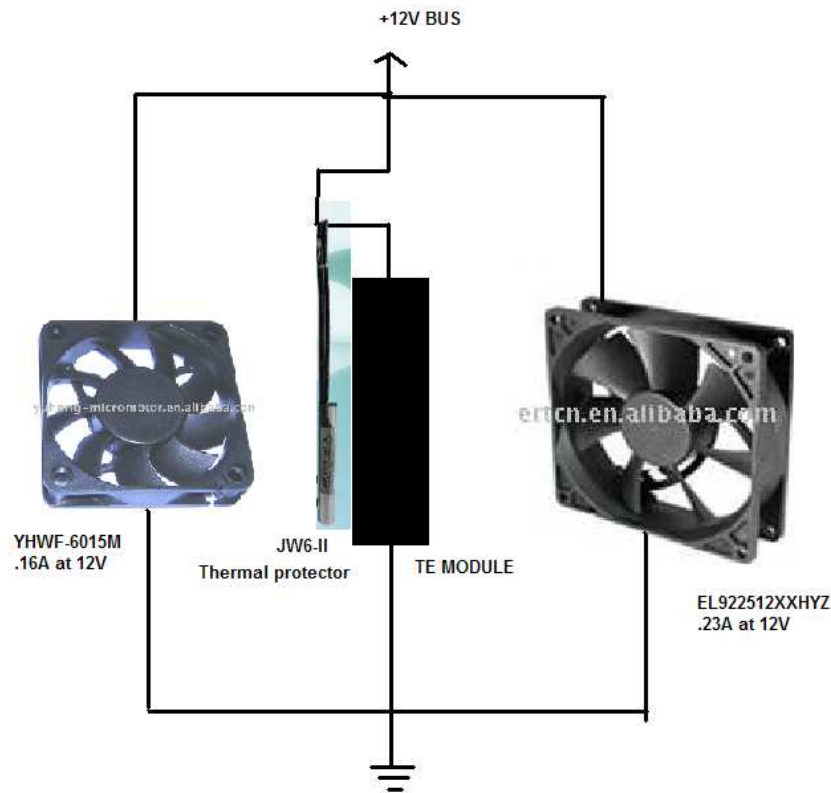
One of the devices on the primary list of loads was a Coleman thermoelectric cooler as shown in Figure 2.5.



**Figure 2.5: Coleman Thermoelectric Cooler [11]**

The thermoelectric cooler uses a device called a Peltier module, which uses semiconductor properties to remove or add heat from the cooler [11].

In order to understand how the unit functioned, a circuit trace was performed. As seen in Figure 2.6 the device contains is a T.E. module, two fans [12] (attached to heat sinks), and in series with the T.E. module is a thermal protector [13] which open circuits at approximately 70-150°C.

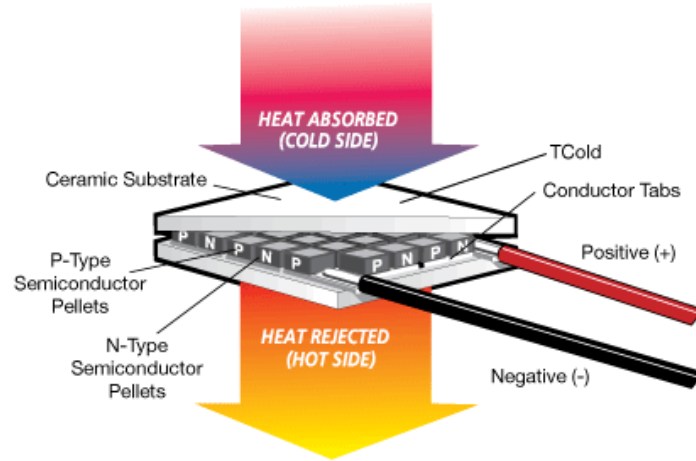


**Figure 2.6: Thermoelectric Cooler Circuit Diagram**

According to Tellurex, a company specializing in thermoelectric modules, the thermal protector is necessary to protect the thermoelectric elements from reaching a temperature which can damage the solder joints or internal structures of the module itself [14].

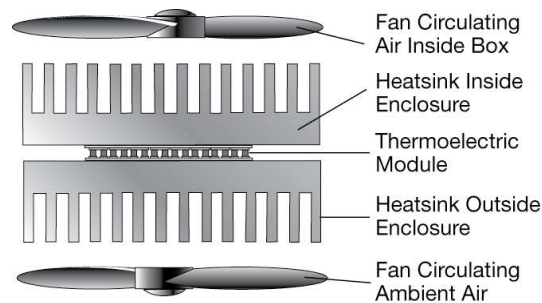
The thermoelectric effect uses specialized materials that when a current is passed through them, one side absorbs heat, while the other side releases heat [14]. The thermoelectric effect can therefore be used for either heating or cooling. Shown in Figure 2.7 is a diagram of the phenomenon. Another application of the technology is thermoelectric power where a temperature differential causes power to be produced from the module [15].





**Figure 2.7: T.E. Effect Diagram [14]**

It is important to note that for the particular application the thermoelectric cooler can only cool to a maximum of 25°C below the ambient temperature [11]. The cooling limit is primarily based on the current limits of the modules. Most typical thermoelectric coolers use a fan-heat sink combination as shown in Figure 2.8. The cooler, fan, and heat sink combination is common among cooler designs and the particular configuration matches the Coleman cooler.



**Figure 2.8: Typical T.E. Cooler Setup [14]**

## **Chapter 3: Enhanced SuPER MATLAB Simulink Model**

### **3.0 Chapter Overview**

This chapter will cover the step by step development of the new SuPER system Simulink model and all of its associated models and sub-models. Also, included in this chapter are the overall design goals for the simulation.

### **3.1 Plans for Enhanced Simulation**

Prior to beginning work on the simulation, the goals were to do three primary things: first, to become familiar with the existing simulation; next, to evaluate the existing model performance and behavior; and finally, to make suggestions for improvement or addition to the existing model.

After reviewing the simulation it was decided that the new simulation should be more modular. In order to modularize the simulation a tool called “create subsystem” was utilized. As shown in Figure 3.1, a very complex system can be condensed into a one modular block. These modular blocks allow for the model to be viewed in an intuitive way. However, if more details are required of a particular subsystem block, then those details can be viewed by double clicking on the block and viewing the different levels of abstraction.

Furthermore, approaching the design in this manner allowed creation of individual subsystem models, outside of the main Simulink model. After testing and behavior verification, the subsystem model could be easily placed into the primary model. The advantage of creating a model outside of the main model is that testing and simulation speed can be increased dramatically. The increase in speed during testing is

due to the fact that there are fewer variables for MATLAB to calculate. Also, a very large step size could be selected depending on the subsystem and this will also speed up the simulation.

To give a better idea of the different layers of complexity, the overall simulation has approximately 153 subsystem blocks to do various calculations, and each simulation (7 days) takes approximately 13 minutes. If an error were to occur in a newly created block, then the fix would have to be made and the simulation would need to be re-run (again taking 13 minutes). If a block is tested and built outside of the overall model, then the model can be run in a couple of seconds and in a much more controlled setting. Shown in Figure 3.1 is the diagram of the new simulation model.

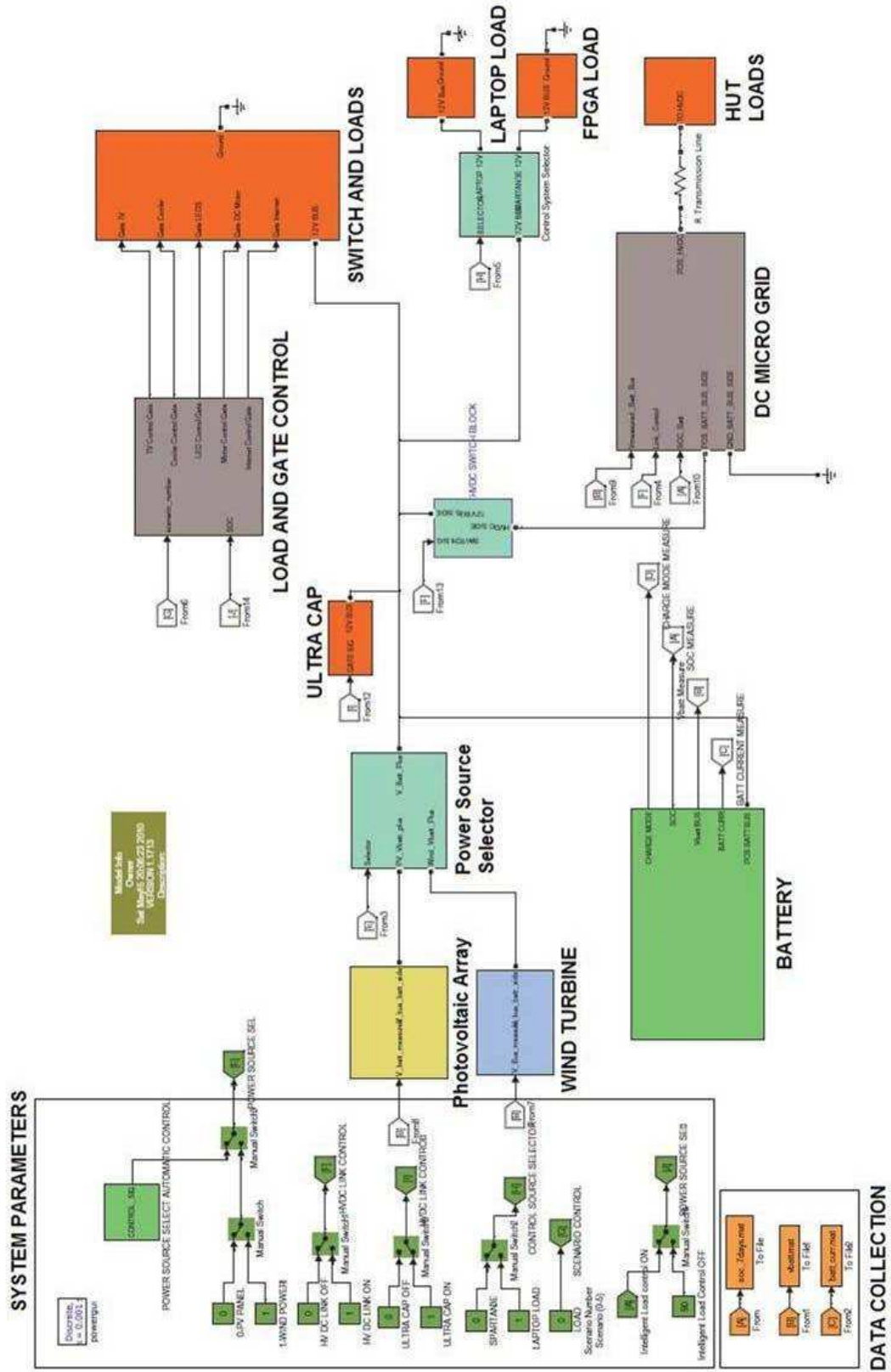


Figure 3.1: Enhanced Simulink Simulation Model

One of the primary requirements in developing a model was that the simulation time had to be fast. The goal was set for the simulation to take under a half an hour to simulate a full seven days worth of data. Previously the simulation took approximately 27 minutes to simulate 2 days worth of data [1]. If future users want to use the simulation to test ideas about system performance, then it would be advantageous to have the simulation run quickly. For example, if the simulation results differ from the experimental data, then the simulation variables can be adjusted, and the simulation output can be quickly re-checked with the experimental data.

## **3.2 Simulink Models**

### **3.2.1 Battery Model**

#### **Speed and Accuracy of the Simulation for the Battery**

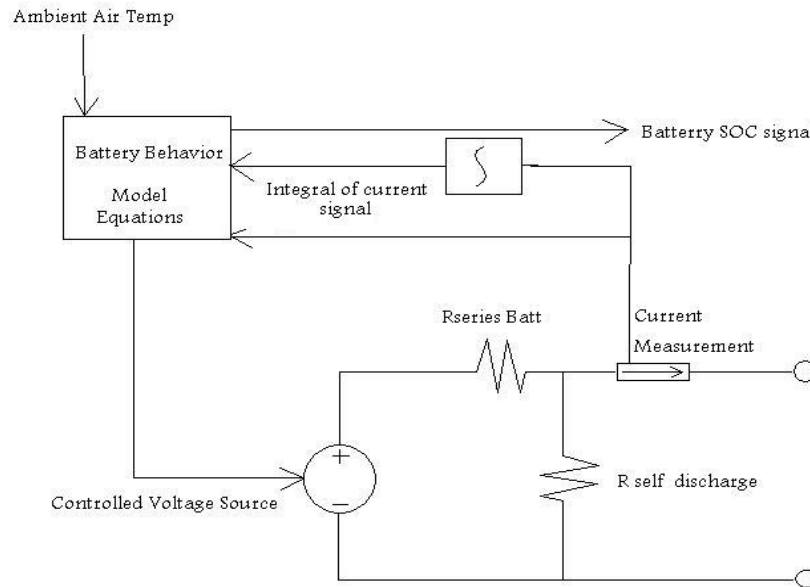
When dealing with battery models the main concession is between detail of the simulation and the speed of the simulation. Primarily the accuracy of the battery model was maintained by using electrochemistry equations and previous experimental data [1]. The speed of the model was preserved by limiting the number of differential elements within the model.

#### **Battery Specifications**

The battery specifications were based on the datasheet as provided by the manufacturer [16]. The primary datasheet item used in the battery model was the battery capacity at 100%. In this case the capacity was 108Ah.

## Overall Battery Model

The overall battery model is a two terminal device that uses a controlled voltage source, current monitor, and an S-function control block to effectively model the given battery. Shown in Figure 3.2 is a basic diagram of this model.



**Figure 3.2: Basic Battery Model**

The self discharge characteristics can be modeled as a resistor because the discharge current is linear in nature. The series battery resistance is primarily made of electrode and terminal resistances which do not change [3]. The series battery resistance was previously experimentally determined [1].

## Basic Charge and Discharge Equations

The discharge equation is based on an adapted form of the Nernst equation (2) as specified in the background section. The only modification was adding in a term to account for the range where the voltage output decays exponentially. The battery output voltage equation used is shown in (12).

$$V = V_{oc} - \left( \frac{Kt_{dis}CT}{C - I_s} \right) + V_e e^{(-Kc_{dis} I_s)} \quad (12)$$

$V_{oc}$  – Open circuit voltage of battery(V)

$V_e$  – Exponential battery value(V)

$Kt_{dis}$  – Discharge conversion constant of the battery  $\left( \frac{V}{^\circ C} \right)$

$Kc_{dis}$  – Current curve constant of the battery  $((A \cdot \min)^{-1})$

$C$  – Capacity of the battery (A · min)

$T$  – Temperature of the battery( $^\circ C$ )

$I_s$  – Integral of the current exiting the battery (A · min)

The charge equations (13) were primarily based on the mathematical relationships as found in Hans Bode [3]. When a battery is charging there are two parts to the charge curve, the first charging portion is usually exponential in nature and this occurs for cases where the state of charge (SOC) of the battery is below 50%. Above 50% the voltage versus time curve takes on a logarithmic relationship. This relationship is primarily due to the charge acceptance rate of the battery and the gassing voltage [3].

$$V = Kt_{\log} \log \left( \frac{\frac{K}{Kt_{\text{dis}}CT}}{\frac{C - I_s}{C - I_s}} \right) \quad (V) \quad \text{SOC} > 50\% \quad (13)$$

$$V = Kt_{\text{exp}} e^{\left( \frac{\frac{K}{Kt_{\text{dis}}CT}}{\frac{C - I_s}{C - I_s}} \right)} \quad (V) \quad \text{SOC} < 50\%$$

$Kt_{\text{exp}}$  – Exponential charging voltage ( $V$ )

$Kt_{\log}$  – Logarithmic charging voltage ( $V$ )

$K$  – Transition voltage ( $V$ )

### The State of Charge Equation (SOC)

The SOC equation (14) is based on how much current is sent into the battery over time. The value is related to the maximum capacity of the battery in amp minutes.

Generally the capacity is given in amp hours, but since the simulation input data is updated every minute, it is more convenient to use these units.

$$\text{SOC} = \frac{K_{\text{SOC}}}{\left( \frac{Kt_{\text{dis}}CT}{C - I_s} \right)} \times 100 \quad (14)$$

$K_{\text{SOC}}$  – SOC conversion voltage ( $V$ )

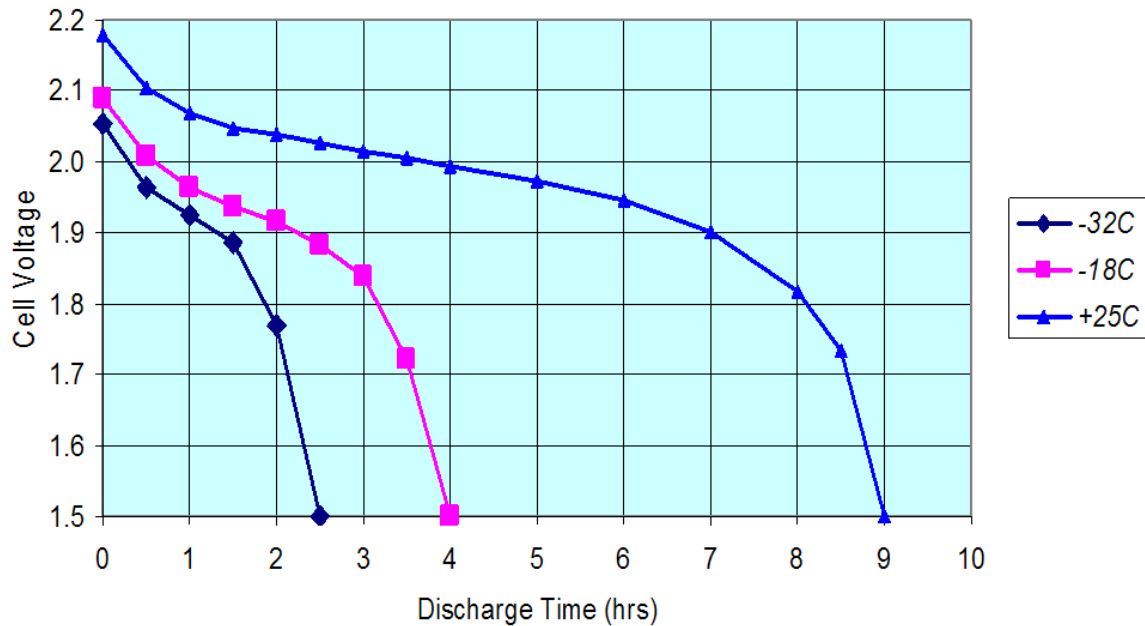
SOC – State of charge (%)

### Experimental Battery Discharge/Charge Curves

All of the constants were empirically determined using data taken by Sheffield [1]. In other words, the constants were adjusted until the experimental discharge data



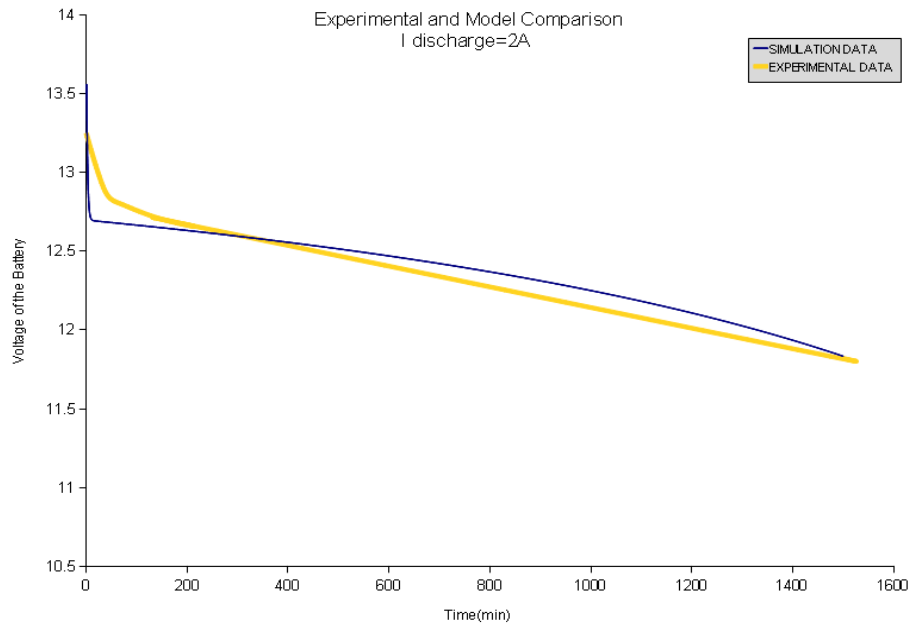
matched the model results. The graphs of the experimental battery data can be found in Figures 3.4, 3.5, and 3.6. One important item to note is that while temperature effects are included in this model, a temperature controlled chamber for the battery is necessary to obtain the data to make a more accurate model. The temperature effects in the model only use the temperature curves as determined via battery experiments (see Figure 3.3). Shown in Appendix A.1 is the C-code written in S-functions to implement the battery model.



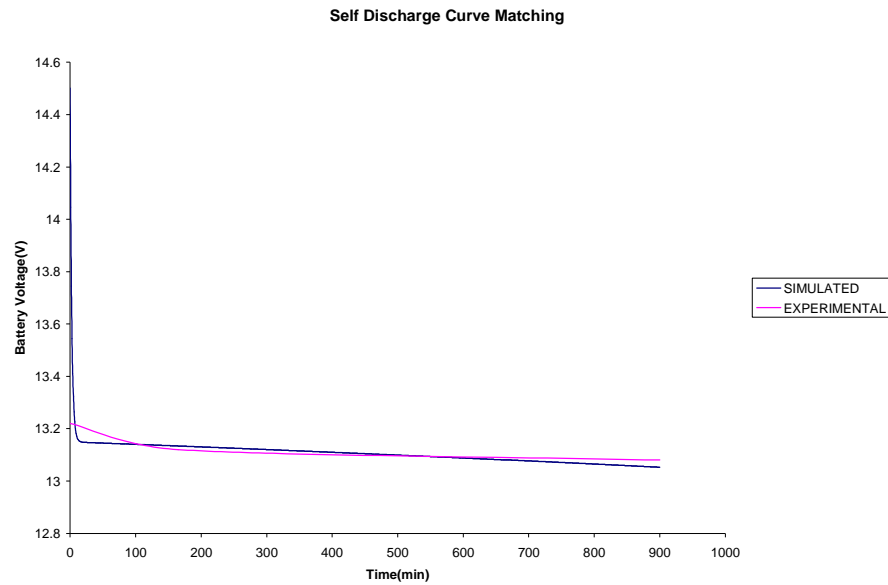
**Figure 3.3: Temperature Relationship Graphs [17]**

### **Battery Data Fitting Equations to the Experimental Data**

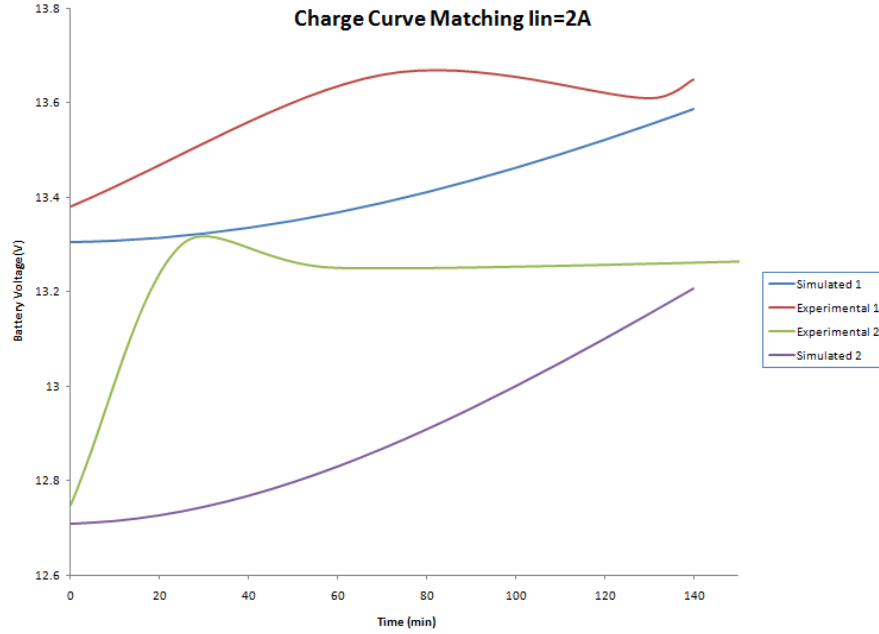
In order to fit the theoretical equations to the experimental data the simulation was run for the exact duration as the current draw/charge experimental data set. The battery constants were then adjusted until the curves matched. Shown in Figure 3.4, Figure 3.5, and Figure 3.6 are plots showing how well the experimental and simulated discharging data compared.



**Figure 3.4: Battery Discharge Data [1]**



**Figure 3.5: Battery Self Discharge Modeling [1]**



**Figure 3.6: Battery Charge Data [1]**

The new battery model is validated by the experimental data. In the battery discharge case the simulation data and experimental data show a maximum difference of .12%. Similarly, the battery charge case simulation data shows a maximum error of 3% with respect to the experimental data.

### 3.2.2 DC Microgrid Modeling

The idea of the DC microgrid came about from conversations with Dr. Robert Van Buskirk [18]. He wanted to use 12V to distribute the power. His reasoning was that since the loads he wanted to supply were low power (LEDs) then the line losses would be low anyways. However, after some discussion with the SuPER group, it was agreed that the best way to send this power would be to increase the voltage. The line losses for even small distances would be too costly in a small scale system like the SuPER project.

### **Length of the Transmission Line.**

The next consideration in modeling a DC microgrid is the distances the system could be expected to transmit power. Judging from the pictures of a typical African village as shown in Figure 3.6 and discussion with Dr. Robert Van Buskirk [18], it was decided that the typical transmission length should be  $\frac{1}{4}$  km from the SuPER system to any given hut. The transmission line length will allow a lumped parameter model of the load side of the microgrid similar to Figure 3.7.

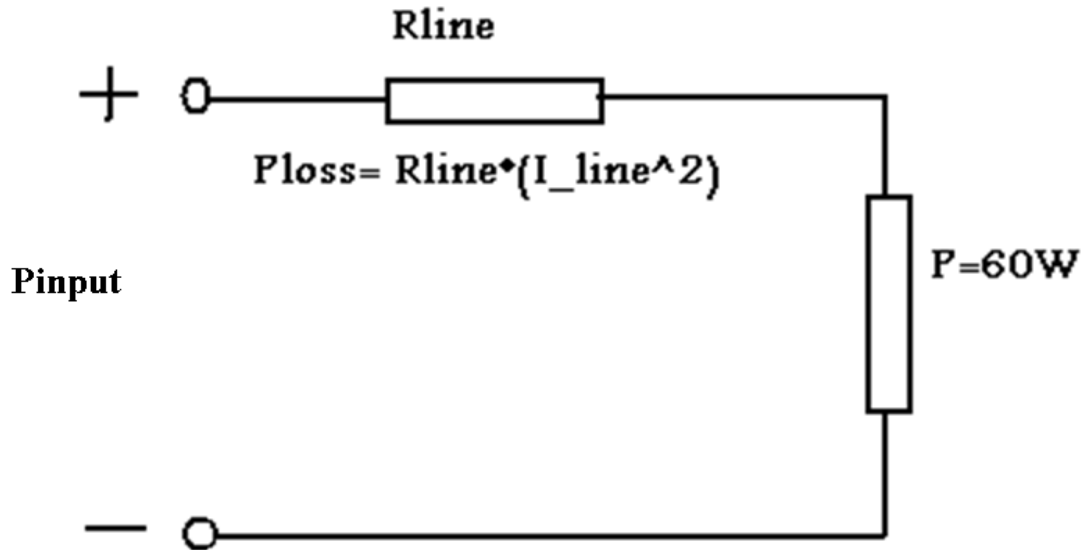


**Figure 3.7: Photo of an African Village [19]**

### **Efficiency Considerations**

The first item of concern in designing a distributed power system is efficiency. The primary loss in efficiency in most transmission systems is line losses. Shown in

Figure 3.7 is an equivalent circuit for a simple transmission system. As shown in Table 3.1 the line losses become significant as the load voltage is lowered.



**Figure 3.8: Efficiency Example**

**Table 3.1: Efficiency Table Example**

<b>Wire gauge</b>	23 (66 ohms/km)					
<b>Distance(Km)</b>	.25					
<b>Vload</b>	<b>Pload(W)</b>	<b>Rline(Ohms)</b>	<b>I_line(A)</b>	<b>Ploss(W)</b>	<b>Pinput(W)</b>	<b>Efficiency (%)</b>
12V	60	16.5	5	412.5	472.5	12.7
160V	60	16.5	0.375	2.32	62.32	96.27

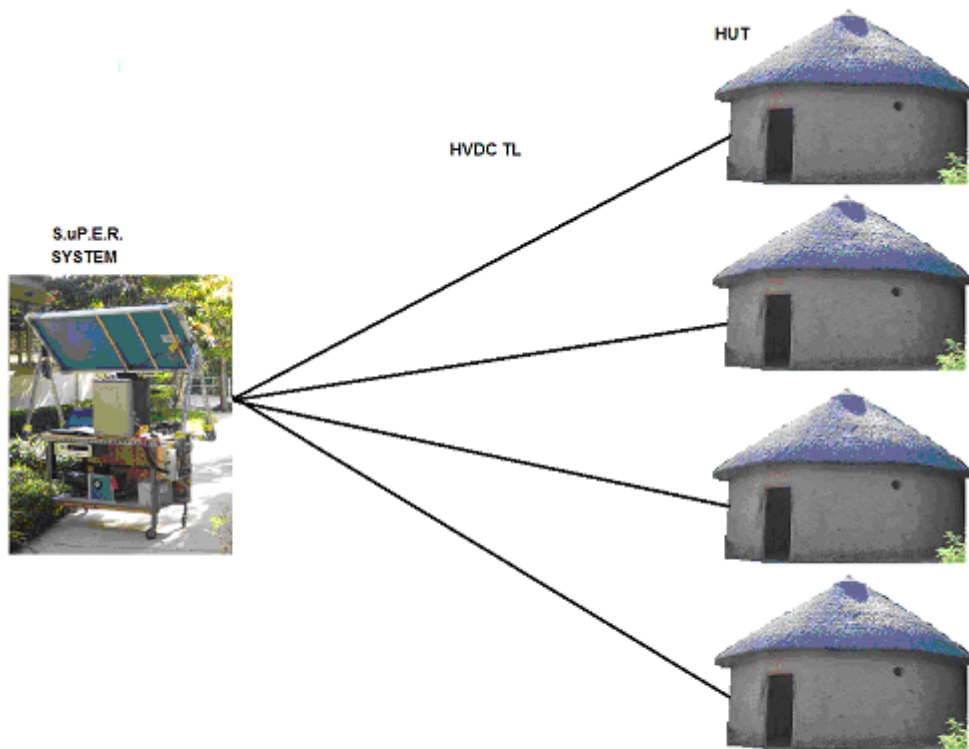
As shown in the example, the best way to increase the system efficiency is to increase the transmission voltage. Of course another way is to select a wire with a lower resistance. However, wires with lower resistance are often thicker, heavier, and cost more. It is a goal of the system to minimize costs and to make implementation easy; therefore increasing the system voltage is the best option.

## **Microgrid Transmission Voltage Selection**

To select an appropriate transmission voltage several things must be considered which include: cost, the available hardware, and what type of loads are expected for the transmission lines to carry. Previous research showed that the ISS is an independent DC microgrid [8]. The ISS has PV panels along with DC-DC converters that step up and down the voltages for transmission and use throughout the station. Since the ISS used 160V DC for a transmission level voltage, then the microgrid system could use the same voltage and achieve similar power savings [8]. Increasing the voltage from 12V DC to 160V DC would have a significant effect by reducing the power losses in the transmission lines. Most home wiring is designed to handle 120Vrms (169Volts peak), therefore setting the transmission voltage to 160V DC means that it would be compatible with existing off-the-shelf hardware.

## **Microgrid Typical Estimated Loads**

Another consideration for the microgrid is the expected system load. In a meeting with Dr. Van Buskirk [18], the expected loads were identified as a cell phone charger, a radio, and LED lighting. Additionally step down DC-DC converters would be needed to be added to bring the 160V DC back to 12V DC at the individual huts. All the loads must be able to operate at 12V DC since that will be the distribution voltage. Shown in Figure 3.9 is a diagram of the structure.



**Figure 3.9: Microgrid Transmission Diagram**

### **Cell Phone Charger**

In order to give a realistic value of how much power consumption could be expected from a cell phone charger a typical charger [20] was selected as shown in Figure 3.10:



**Figure 3.10: Typical Cell Phone Charger**



**Figure 3.11: Power Consumption View of a Cell Phone Charger [20]**

According to the specification sheet the maximum power draw was listed at 2.5W per cell phone charger. For the model, it is assumed that there is only one cell phone charger per household. While, it is true that the cell phone charger may not always be in use (i.e., the cell phone is charged), to be conservative the charger will be set to operate about two hours a day.



## Radio

A rechargeable 12V radio [21] was selected for the radio model. The radio shown in Figure 3.12 has the DC power input listed at 12V at 500mA. Therefore the radio being charged or used will consume 6W.



**Figure 3.12: Sangean Radio [21]**

## LED lighting

One of the most important items to consider as a load is lighting. LED lighting was chosen due to the low power consumption, long life, and good light output. Choosing a light source that consumed a lot of power would not be sustainable with the low power SuPER system.

To pick a suitable LED light, the light source would have to operate directly from 12V DC and it would ideally give the light output of a 60W incandescent light bulb. A suitable match was found [22] and the specifications are shown in Figure 3.12. As shown in the data sheet, this light only consumes 5 watts of power, and yet it outputs approximately the same light as a 60W incandescent bulb. The power savings with LED technology is phenomenal, not to mention that since LED bulbs are solid state, therefore they will last a very long time. The LED lighting stays consistent with the SuPER

project and its overall goal of a lifespan 20 years [23]. The model will assume that each hut will have one bulb.



EO EL16 A - 51W3, a 10 degree beam width MR16 in White. GU5.3 Base. This MR16 has a single 5 watt Edison Opto LED made from the Cree die. Light color is excellent and clear.

- LED: 1 piece 5 Watt Edison Opto LED
  - View Angle: 10 degree view angle or beam width
  - Voltage: 12 volt DC and AC
  - Lux: 1300@ 1 meter
  - CRI: 80
  - Light Output: Estimated at 60+ Watts
  - Color Temp: 6300-7000 Kelvin White
  - Power Consumption: 5.0 watts
  - Operating Temperature: 40° C, or 104° F.
  - Dimensions: 1.97" x 1.93"
  - Not dimmable
  - Certification: CE Approved; RoHS Compliant
  -
- Warranty: 1 year from purchase date

**Figure 3.13: LED Light Bulb [22]**

### Estimated Lumped Load

In order to provide a typical load scenario all of the loads from one hut will be summed up.

$$P_{consumed\ 1\ Hut} = P_{LEDs} + P_{Radio} + P_{cell\ Phone\ charger} \quad (15)$$

$$P_{LEDs} = 5W$$

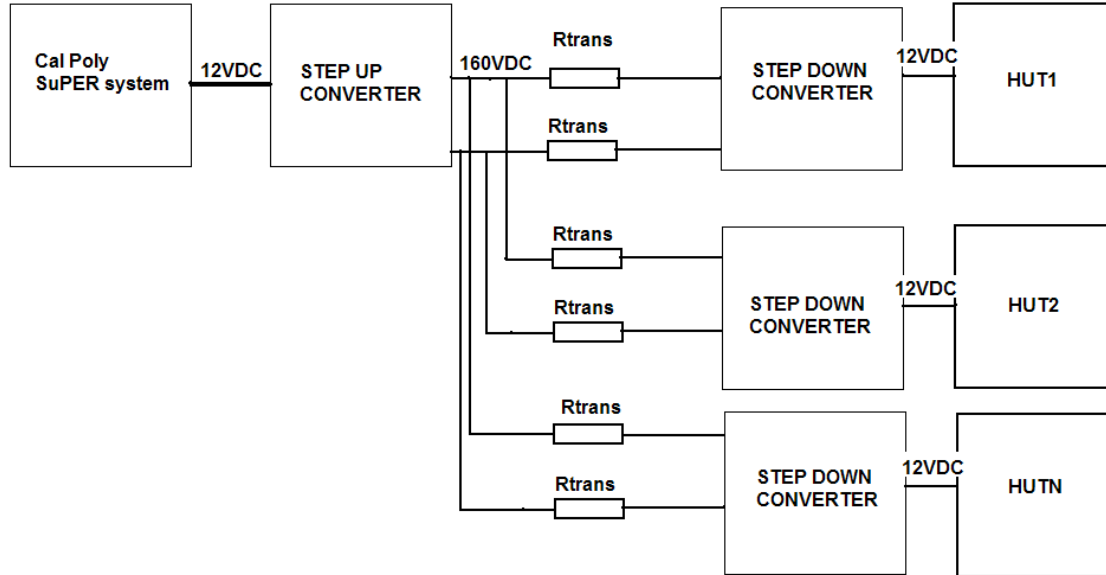
$$P_{Radio} = 6W$$

$$P_{cell\ Phone\ charger} = 2.5W$$

$$P_{consumed\ 1\ Hut} = 5W + 6W + 2.5W = 13.5W / hut$$

The maximum power consumption of the microgrid was arbitrarily set to be 60W, which is half of the PV panels maximum output. Line losses and the efficiency of the DC-DC converters (both step up and step down) must also be considered. The resistance

of the transmission line was found based upon a typical wire gauge chart [24]. The entire equivalent circuit diagram can be seen in Figure 3.14:



**Figure 3.14: Lumped Load Diagram**

In order to solve for the actual amount of huts that can be supplied given the proposed configuration, the following equations (16) are presented for the analysis.

Assume :

(16)

a) Wire gauge : 23 gauge ( $66 \frac{\Omega}{\text{Km}}$ ) [24]

b)  $R_{\text{line}} = 16.5 \Omega$  (Assume .25Km distance)

c)  $P_{\text{load}} = 13.5 \frac{W}{\text{hut}} \Rightarrow R_{\text{load}} = \frac{(160V - R_{\text{line}}I)^2}{13.5W}$

d)  $P_{\text{transmission}} = R_{\text{eq}} I^2$

e)  $R_{\text{eq}} = \frac{1}{G_{\text{eq}}}$

f)  $G_{\text{eq}} = G_B + G_B + \dots + G = nG_B \Rightarrow R_{\text{eq}} = \frac{1}{nG_B}$

g)  $P_{\text{upconv}} = .05P_{\text{req}}$

h)  $P_{\text{downconv}} = .05P_{\text{load}} = .675W$

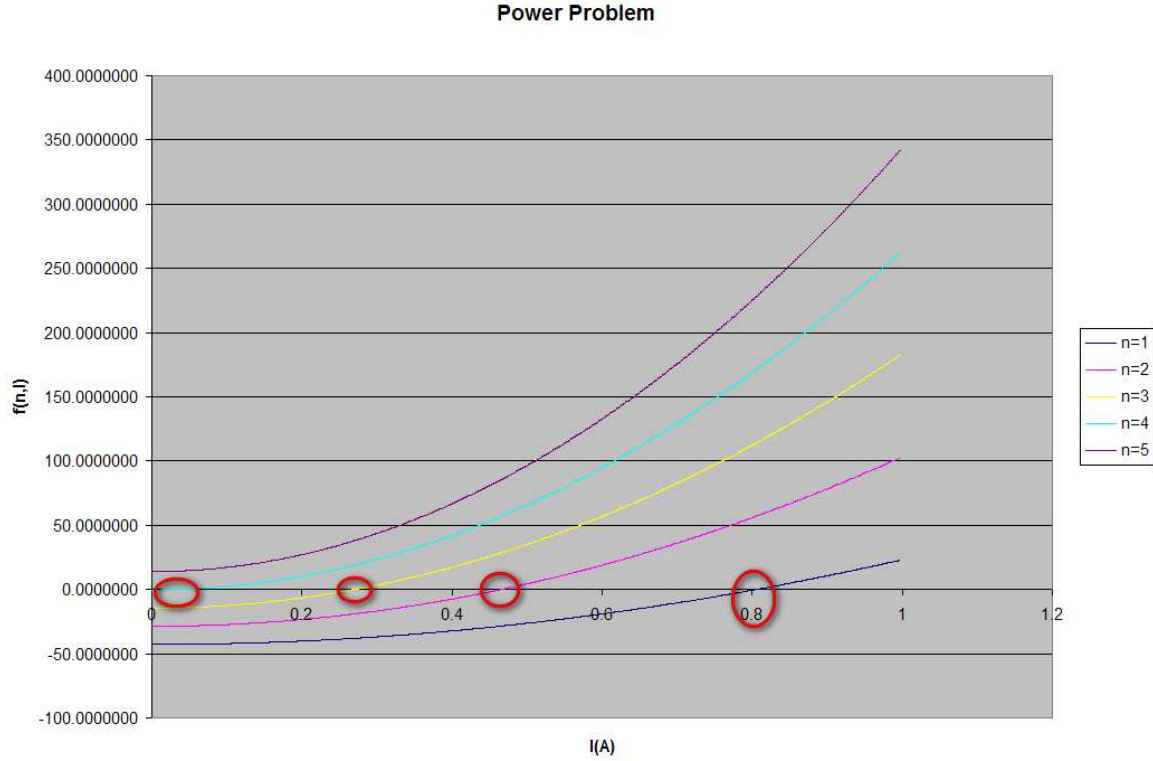
$P_{\text{req}} = n(P_{\text{load}} + P_{\text{downconv}}) + P_{\text{transmission}} + P_{\text{upconv}}$

$$P_{\text{req}} = n(13.5W + .675W) + \frac{1}{n \left( \frac{1}{\frac{(160V - (16.5 \Omega)I)^2}{13.5W} + 16.5 \Omega} \right)} I^2 + 3W$$

$$57W = n(14.175W) + \frac{I^2}{n} \left[ \frac{(160V - (16.5 \Omega)I)^2}{13.5W} + 16.5 \Omega \right]$$

$$\boxed{f(n, I) = n(14.175W) + \frac{I^2}{n} \left[ \frac{(160V - (16.5 \Omega)I)^2}{13.5W} + 16.5 \Omega \right] - 57W}$$

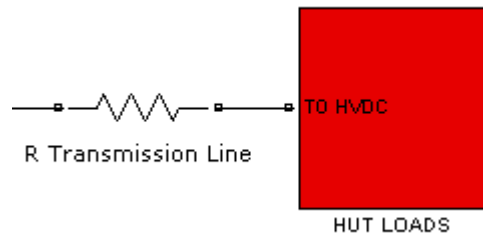
The plot of the function as shown in Figure 3.15 shows that the last real value is the most optimal solution (lowest current). Therefore the maximum number of loads that can be supplied is 4 huts.



**Figure 3.15: Hut Power Consumption Graph**

### Microgrid Load and Load Control Models

Using the existing framework from the primary loads and load control, a block was created which essentially does the same thing for the hut loads. The block cycles the loads depending on the time of day. The load schedule chosen can be seen in Figure 4.4. The code which controls this block is listed in Appendix A.3 Hut Scenario Load Control Block S-Function Code. The block diagram of the hut loads is shown in Figure 3.16. The only input into this block is the positive terminal of the high voltage DC link.



**Figure 3.16: Hut Loads Block Diagram**

### **DC-DC Converter**

In order to build a realistic model, the transient effects of the DC-DC converter must be modeled. Also some of the primary concerns in selecting the DC-DC converter are:

1. It must be able to handle switching voltages at a minimum of 160V DC.
2. The converter must have short circuit protection for safety reasons.
3. The converter must be as efficient as possible over all load ranges.
4. The converter must automatically shut off if the battery state of charge drops below a predetermined level.

When it comes to DC-DC converters, there is a multitude of types for many different purposes. Generally DC-DC converters are divided into two types, either step up (Boost) or step down (Buck). Step up takes the input voltage and steps it up to a higher voltage at the output, while step down does the opposite [25]. First, the voltage will need to be stepped up to transmit it and then the voltage will be stepped down to a usable level at the load side.

Now, the class of DC-DC converter that will be modeled for the microgrid is called a resonant converter. The resonant converter circuit uses a LC resonant behavior to force the voltage or current value to zero during switching transitions [25]. To be consistent with the 25 year MTBF a resonant converter is necessary to lower stresses on the switching element.

The zero voltage switching (ZVS) DC-DC converter as shown in Figure 3.17 is operated such that every time the switch opens, there is essentially zero voltage across it. This relationship shown in Figure 3.17 allows stresses on the switch to be lowered, especially at high voltages. Also, the zero voltage switching is applicable to many switching types (Buck, Boost and Flyback) [25].

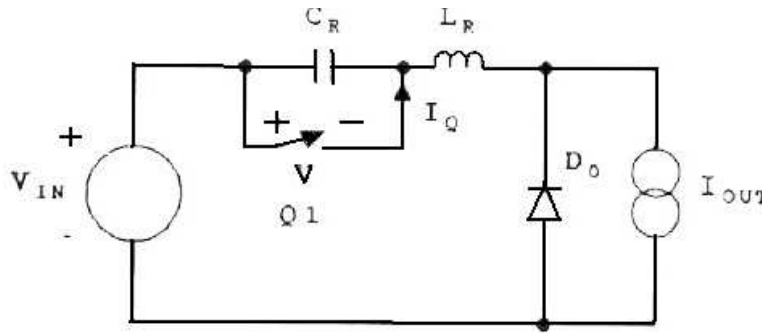


Figure 3.17: Buck ZVS Circuit [25]

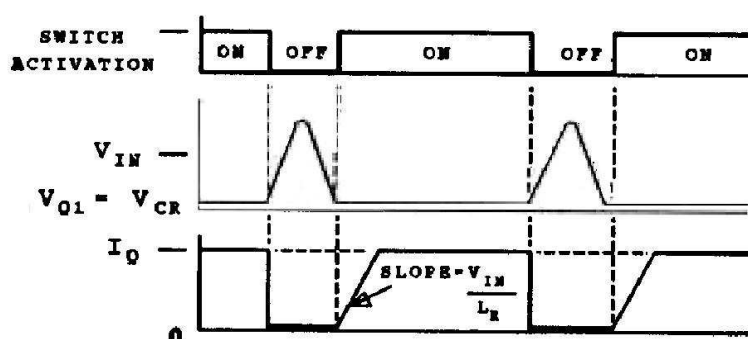
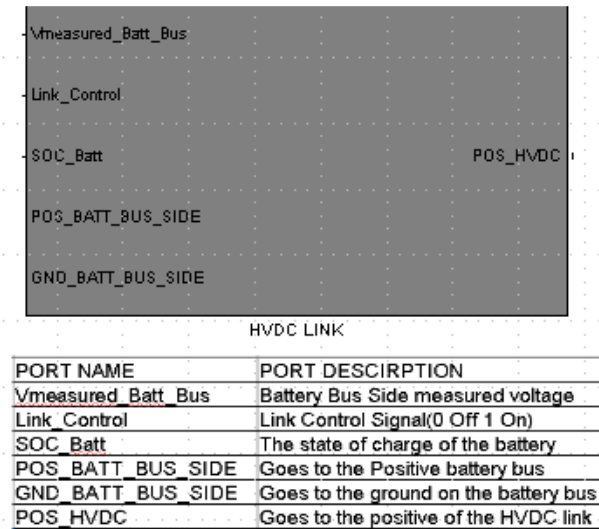


Figure 3.18: ZVS Waveforms [25]

## MATLAB Implementation of the Microgrid

In order to implement the DC microgrid in MATLAB the current simulation speed was to be maintained, while still representing a realistic load as seen from the generation side. The MATLAB block had to be versatile so that if a different lumped load were used, then the model could reflect those changes. Additionally, a link control was necessary to allow for manual (ON-OFF) control of the microgrid. The MATLAB block that was created is shown in Figure 3.18.



**Figure 3.19: DC Microgrid Block Diagram**

As shown in Figure 3.20 the transmission line and load resistance are lumped into a variable resistor. To change the hut loading, one would change the resistance values on the hut side and then the converter would reflect the increased loading on the battery bus side. The ability to perform this dual side loading was made possible by a variable resistor on the battery bus side that is controlled via C-code.





safeguards focus on excessive current draw and depleting the battery excessively.

Similar to the intelligent load control principles, the DC microgrid is set to shutdown once the SOC drops below 80%. Additionally, the microgrid is configured to shutdown if a link shutdown condition is asserted. Listed below are the link shutdown conditions:

- SOC of the battery drops below 80%
- The Power consumed by the link exceeds the power consumptions set point (Currently set to 70W).
- A fault condition exists indicated by an above average current (Currently set to 500mA).
- Link Shutdown terminal is asserted Low (0) (If terminal is high(1) then the microgrid will function normally).

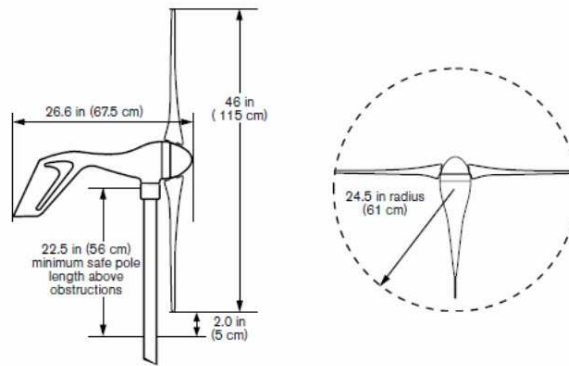
### **3.2.3 Wind Turbine Design and Modeling**

Initially, the Cal Poly SuPER system did not have a wind turbine incorporated into the system. However, adding additional generation capability to the system would add another renewable energy power source that would allow for increased operation and use of the entire system. During times of low sunlight, the system could be used when wind was present.

#### **Selection of a Suitable Turbine**

When deciding how large a wind turbine to model, the power generation of the current PV system was considered. After further review of the data sheet [26] and experimental data [1] it was found that the photovoltaic system currently put out 120W of power at peak insolation. If the purpose is to supplement the PV power then the wind

turbine must at least match the power generation of the existing solar panel (for occasions when there was little or no sun). Wind turbines are not always running at their peak power generation; therefore to be conservative the assumption was made that on average the wind turbine would only output 30% of the maximum rated power at any time. Consequently a 400W wind turbine was chosen. The turbine must also be a commercially available design, easy to implement, rugged, and not too big. The Air X wind turbine [27] was chosen because it met all of the requirements. As shown in Figure 3.21 and 3.21, the turbine is rugged and built for even marine environments.



**Figure 3.21: Air X Wind Turbine Dimensions [28]**



**Figure 3.22: Air X Wind Turbine Picture [28]**

The turbine selected also has additional features such as having a built in charge controller and an automated braking feature that engages when the wind speed exceeds a certain speed to prevent mechanical damage to the rotor.

### **Wind Turbine control**

Fortunately, all of wind turbine control for the Air X turbine is controlled by an internal microprocessor that performs all of the voltage regulation and rotor speed control.

### **Output Voltage Control**

As stated in the user manual [28], the output voltage regulator can be set according to the data in Table 3.2.

**Table 3.2: Air X Wind Turbine Voltage Regulation Table**

<b><i>AIR-X Adjustable Voltage Range</i></b>	
<b><i>12v</i></b>	<b><i>13.6v – 17.0v preset to 14.1v</i></b>
<b><i>24v</i></b>	<b><i>27.2v – 34.0v preset to 28.2v</i></b>

The turbine microcontroller will regulate the output voltage to the values in Table 3.2 as long as the power is available from the turbine itself. The turbine uses DC voltages and therefore the system is consistent with the rest of the SuPER system.

According to the manual [28], when the battery reaches the upper charge set point the microprocessor automatically sends a signal to start slowing down the rotor and stop charging the batteries. Once the battery voltage drops below the lower set point, the microprocessor will allow the rotor to speed up again and charge the batteries.

## Rotor Speed Control

According to the user manual [28] the rotor speed control is accomplished using a microprocessor, a rotor speed sensor, and a peak set point algorithm. The Air X is able to control the rotor speed by increasing the current drawn from the turbine, which makes it harder for the rotor to turn (3 phase permanent Magnet rotor). In doing so, the Air X can effectively regulate and slow down the blades at speeds up to 110mph.

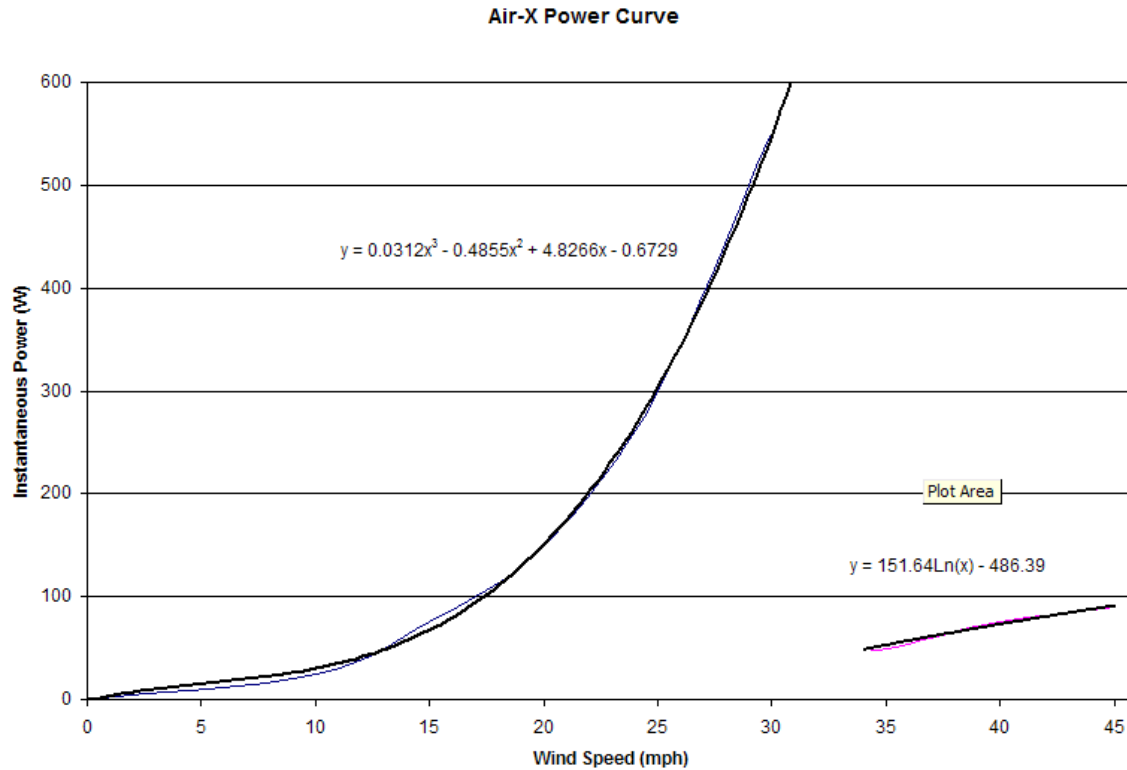
## Power Curve Modeling

In order to accurately represent the wind turbine power output, the data sheet chart [27] shown in Figure 3.23 was utilized.



**Figure 3.23: Air X Wind Turbine Power Output Graph [27]**

A curve of best fit was performed on the wind data and the output power equations were obtained. The equations found were then used directly in the MATLAB model. Figure 3.24 presents the excel chart and the equations of best fit.



**Figure 3.24: Air X Wind Turbine Power Curve Fitting**

### Wind Turbine Resistance Calculations

According to the wind turbine user manual a recommended wire gauge chart (Table 3.3) yields the expected gauges depending on the length of the transmission wire.

**Table 3.3: Air X Wind Turbine Wire Gauge Chart [28]**

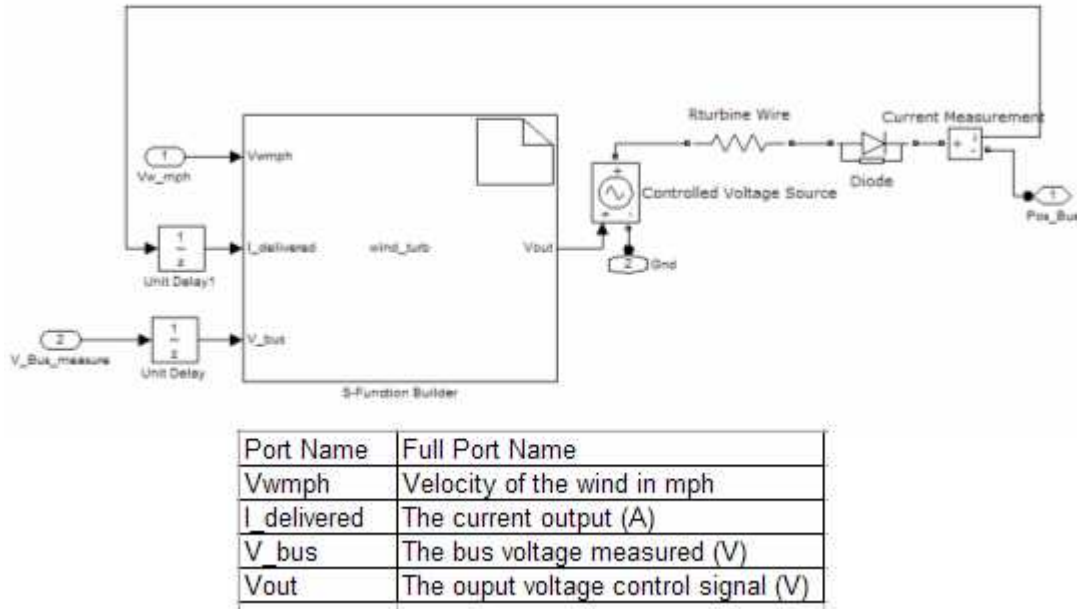
**12V AIR-X: Wire Size, Considering 12mph (5.2 m/s) Wind Average and 95% Energy Transmission Efficiency. Wire Size: AWG / sq mm**

# Turbines	0-30ft	30ft-60ft	60ft-90ft	90ft-150ft	150ft-190ft	190ft-250ft	250ft-310ft	310ft-390ft	390ft-500ft
1	8g/8	6g/13	4g/21	2g/34	1g/53	0g/53	00g/67	000g/85	000g/85
2	6g/13	4g/21	1g/44	00/67	000g/85	0000g/107	*	*	*
3	4g/21	2g/34	0g/53	000g/85	0000g/107	*	*	*	*

It will be assumed that the height the wind turbine will be mounted at is 80 feet. The chart gives a recommended 4 gauge wire size. Referring to a wire gauge chart [24] shows a resistance of approximately .81ohm/Km. Converting this to feet gives .000246ohms/foot. Therefore the resistance will be .01975 ohms for the 80 foot distance.

## Wind Turbine Simulink Block Implementation

To implement all of the controls, power curves, and resistance of the wind turbine, a MATLAB S-Function block was used as shown in Figure 3.24. The code to control the S-function block is shown in Appendix A.4 Wind Turbine S-Function Code.



**Figure 3.25: Wind Turbine Simulink Block**

### Wind Data Input Sources.

Existing studies of wind turbines indicated that in most cases, all of the simulations of wind turbines were driven with data that was not actual data. Most, if not all of the simulations used Monte Carlo Simulations [29], which use random input data that conforms to a certain statistical model of how a system input behaves. Monte Carlo data was not chosen because a real set of data would allow the system to accurately reflect what would really happen for an actual 7 day scenario.

A great deal of wind data had already been recorded by Professor John Ridgely in cooperation with the Cal Poly wind turbine project [30]. The data recorded was provided in a raw format and some processing was done.

### Wind Data Processing

The only necessary data for the model would be the time stamp and the wind speed. However, much more data was provided in the raw data file. Below in Figure 3.26 is an example of a typical data file.

```
^ 20:50:47 12 16 36 0 0 0
^ 20:51:06 18 9 29 0 0 0
^ 20:53:30 14 26 33 12 0 0
# 28 12 16:55:01 24 28 22 24 28 22 -14 -6
^ 20:55:03 11 0 0 0 0 0 12.98V
^ 20:55:13 16 23 33 17 -125 18
^ 20:55:50 12 25 34 18 0 0
^ 20:56:21 18 21 30 2 0 0
^ 20:58:38 14 14 26 0 0 0
# 28 12 17:00:01 23 27 19 23 27 19 -13 -7
^ 21:00:03 11 0 0 0 0 0 12.98V
^ 21:00:24 16 6 19 0 -120 4
^ 21:00:55 12 9 24 0 0 0
^ 21:01:38 18 1 9 0 0 0
^ 21:03:46 14 0 0 0 0 0
# 28 12 17:05:01 24 29 19 24 29 19 -12 -10
^ 21:05:36 16 0 3 0 -74 -98
```

**Figure 3.26: Raw Wind Data File**

The information provided is as follows: First the start character “^”, the time stamp, the sensor number, the average wind speed(mph/knts), the maximum wind speed(mph/knts), the minimum wind speed(mph/knts), cosine(wind direction), sine(wind direction). The length of one day’s worth of data is about 41 pages worth of text information; therefore a computer algorithm was written to pick the necessary data out of these files.



The first algorithm written (see Appendix B section B.1) in VBA grabbed all the data corresponding to a particular sensor number and placed it in the necessary format (see Figure 3.27). All of the data from all four sensors was processed in case different wind turbine heights needed to be evaluated. The files of the one day data sets were combined into one continuous string of the data that corresponded to the seven day scenario. Shown in Appendix B section B.2, is the program that was written.

1	TIME STAMP	AVG WIND SPEED
2	00:06:29	14
3	00:11:45	3
4	00:17:03	3
5	00:22:18	0
6	00:27:33	9
7	00:32:51	13
8	00:38:05	21
9	00:43:21	25
10	00:48:37	33
11	00:53:53	43
12	00:59:10	39
13	01:04:27	40
14	01:09:42	38
15	01:14:57	31
16	01:20:14	22
17	01:25:29	18
18	01:30:45	4
19	01:36:02	10
20	01:41:18	14
21	01:46:33	22
22	01:51:50	17
23	01:57:06	11
24	02:02:21	9
25	02:07:38	17
26	02:12:53	4
27	02:23:25	0
28	02:28:41	0
29	02:39:14	16
30	02:44:29	18
31	02:49:46	32
32	02:55:03	36
33	03:00:18	35
34	03:05:33	17
35	03:10:52	22
36	03:16:05	20

**Figure 3.27: Processed Wind Data**

Finally once the data was in the correct format it was placed into readable format for MATLAB. Originally, the previous model used the lookup table to implement the source data for the PV array [1]. A lookup table with 10,080 data points (7 days worth of data updated every minute) was created; however a memory error was received. Fortunately, the “from file” command was discovered. The “from file” command will

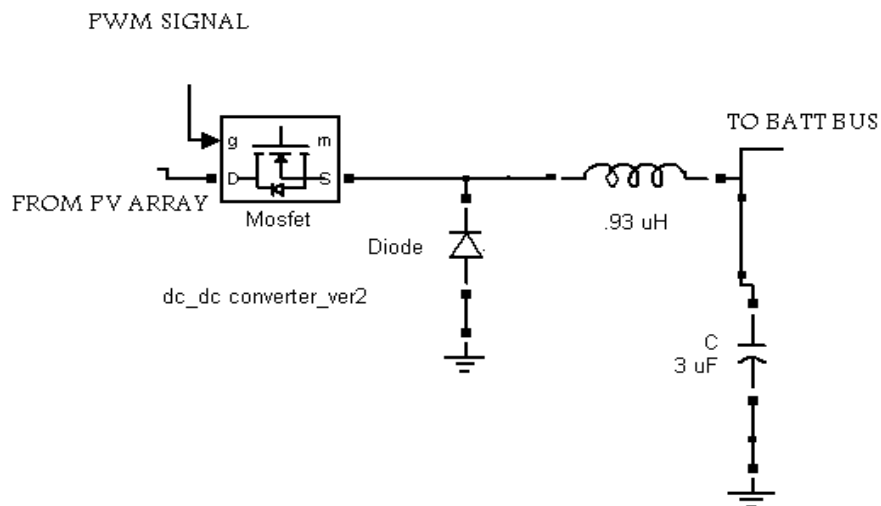
open up a .MAT file and it will step through each row of data at the specified sample rate (see Appendix C Using from file Block for more information). An additional benefit to this function is that the “from file” block has less memory overhead and will therefore aid in speeding up the simulation.

### 3.2.4 New PV DC-DC Converter Model

The development of the new DC-DC converter took three versions. The initial DC-DC converter that was modeled by Sheffield [1] was the MX-60 [31]. However, after work by Witts [32] a new DC-DC converter was implemented. Because a new DC-DC converter was now being used, a new model would have to be created.

#### Old Model

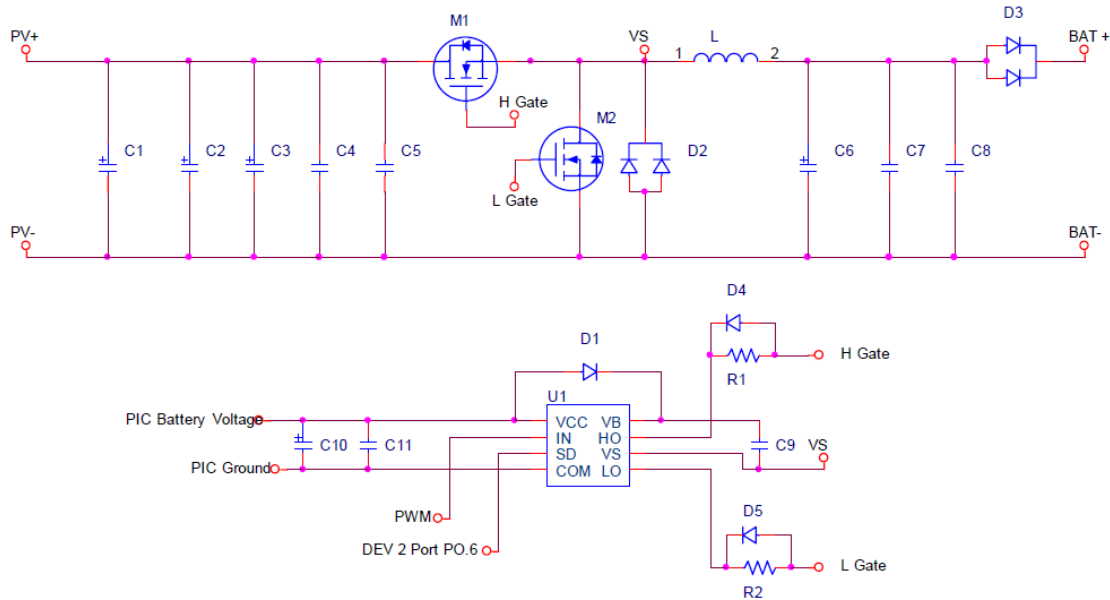
As shown in Figure 3.28 Sheffield [1] created a simple buck DC-DC converter to model the MX-60 DC-DC converter and charge controller.



**Figure 3.28: Old DC-DC Converter Model [1]**

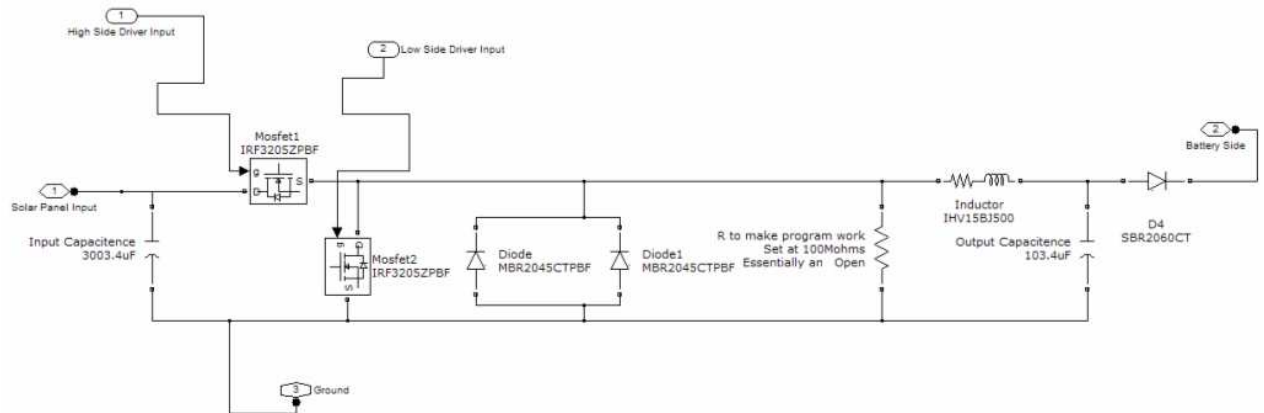
## Attempted New Model

The goal in creating the new model was to replicate exactly what existed with the new DC-DC converter that Witts [32] had created (see Figure 3.29). The direct circuit diagrams were used to re-create a model that matched what existed. Every data sheet and specification of every part was made to match the actual parts used.



**Figure 3.29: Circuit Diagram of New DC-DC Converter [32]**

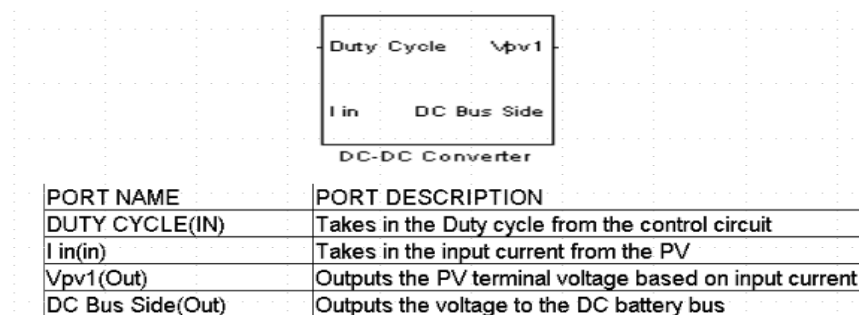
The model that was created (see Figure 3.29) worked exactly as it should have; however, the simulation was significantly slowed down (see Appendix D section D.1) due to the increased number of differential elements (R,L,C) and the required minimum step frequency of 60 KHz. This minimum step frequency was based on the operating frequency of the DC-DC converter which was 30 KHz. In order to run a full seven days worth of data, a much faster DC-DC converter model would have to be created or faster computers would have to be used.



**Figure 3.30: Re-Created DC-DC Converter in Simulink**

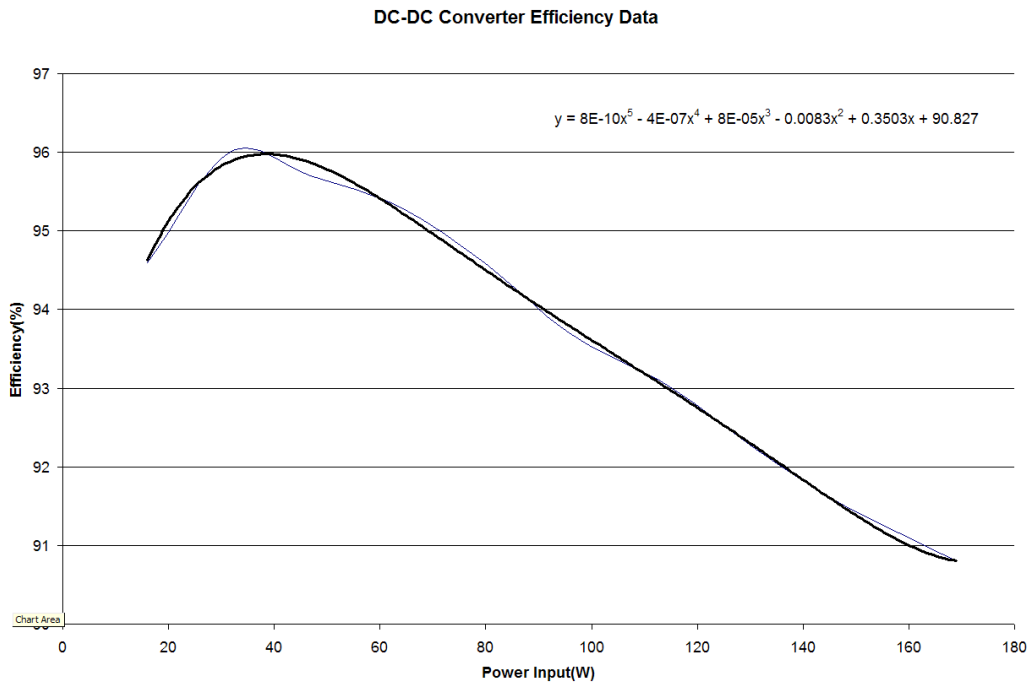
### Final Revision for Improved Speed.

In order to improve the simulation speed it was decided to do away with modeling the transient effects of the DC-DC converter. The minimum required PWM frequency and the use of differential elements are the primary reasons why the simulation time was so large. Therefore the third and final revision of the DC-DC converter was based purely on mathematical relationships instead of differential elements (see Figure 3.31). These changes allowed the simulation speed to be dramatically increased (Appendix D section D.2).



**Figure 3.31: DC-DC Converter Block Diagram**

The block was modeled using an S-Function block coded in C (Appendix A.5 PV DC-DC Converter Control S-Function Block Code). The converter efficiency data experimentally determined by Witts [23] was used to create a curve of best fit for the DC-DC converter. This characteristic allowed the model to represent how power coming from the PV array was being transmitted through the DC-DC converter as seen in Figure 3.32.



**Figure 3.32: DC-DC Converter Efficiency Data [32]**

The primary boost in speed for this model was due to the fact that the step size was no longer dependent on the pulse width modulation frequency of the DC-DC converter [32]. Since the transient effects of the DC-DC converter were not an issue, this meant that the step size could be much lower. Now the only item limiting the step time was the time required to represent the transient effects of the DC motor model. In the

case of the motor the transient effects were important because the start up current required by the motor was significant.


### **3.2.5 PV Control Algorithm Modification**


Initially, the PV control algorithm model was created from the existing C code on the SuPER control laptop (pnopal94khz.C). However, a problem was discovered when running this code in the simulation. After approximately two days of operation the duty cycle being sent to the DC-DC converter model simply went to zero. This essentially meant that even though there was input power available, the system was simply shutting down. Since the system has never been run for that length of time, it is assumed that similar results would occur in the real world.

The PV control algorithm was re-written from the same charging principles used for the MX-60 charge controller [31]. The code is shown in Appendix A.6 PV Charge Control Block S-Function Code. The completed code was tested and verified to work within the simulation.

### **3.2.6 Spartan 3E Control Board (FPGA)**

The proposed Spartan 3E control board was a senior project worked on by Matt Staniszewski [33]. The idea was to replace the existing laptop with a single FPGA. The FPGA would lower the power consumption and size of the control unit. The estimated power consumption of the FPGA was 30mW as shown Figure 3.33.





Device

Part	XC3S100E
Package	TQ144
Grade	Commercial
Process	Typical

Block Summary

Block	Power (W)
CLOCK	0.000
LOGIC	0.000
IO	0.000
BRAM	0.000
DCM	0.000
MULT	0.000

Voltage Source Information

Source	Voltage	Power (W)	I <sub>CC</sub> (A)	I <sub>CCQ</sub> (A)
V <sub>CCINT</sub>	1.2	0.010	0.000	0.008
V <sub>CCALIX</sub>	2.5	0.020	0.000	0.008
V <sub>CCQ 3.3</sub>	3.3	0.000	0.000	0.000
V <sub>CCQ 2.5</sub>	2.5	0.000	0.000	0.000
V <sub>CCQ 1.8</sub>	1.8	0.000	0.000	0.000
V <sub>CCQ 1.5</sub>	1.5	0.000	0.000	0.000
V <sub>CCQ 1.2</sub>	1.2	0.000	0.000	0.000

Thermal Information

Ambient Temp (°C)	25.0
Airflow (LFM)	250
ΘJA (°C/W)	40.5
Custom ΘJA	
Max Ambient (°C)	83.8
Junction Temp(°C)	26.2

Power Summary

Optimization	None
Data	Production
Quiescent(W)	0.030
Dynamic (W)	0.000
Total (W)	0.030

Import from ISE...

Reset to Defaults

Import from XPE...

Set Toggle Rate...

Advanced Options...

Set Default Clock...

Comments

This area is for the user to enter any comments about the design, operating conditions, assumptions or anything of relevance.

[XILINX Power Central on the Web](#)  
[Email Feedback](#)  
Workbook covered under the [license agreement](#)  
© Copyright 1994-2008 Xilinx, Inc. All Rights Reserved

**Figure 3.33: Xilinx Power Estimator Worksheet**

### 3.2.7 Satellite Modem and Wireless Router

A satellite modem was selected to allow for Internet use in any location that had a view of the sky. In this case the Hughes DW7000 satellite modem [34] was selected as seen in Figure 3.34 and Figure 3.35 . In this case the maximum power consumption was found to be 12W.



**Figure 3.34: Direct Way Satellite Modem [34]**



**Figure 3.35: Direct Way Satellite [34]**

A wireless router was selected to save money in wiring and for simplicity. The unit as shown in Figure 3.36 has power consumption of approximately 12W [35].

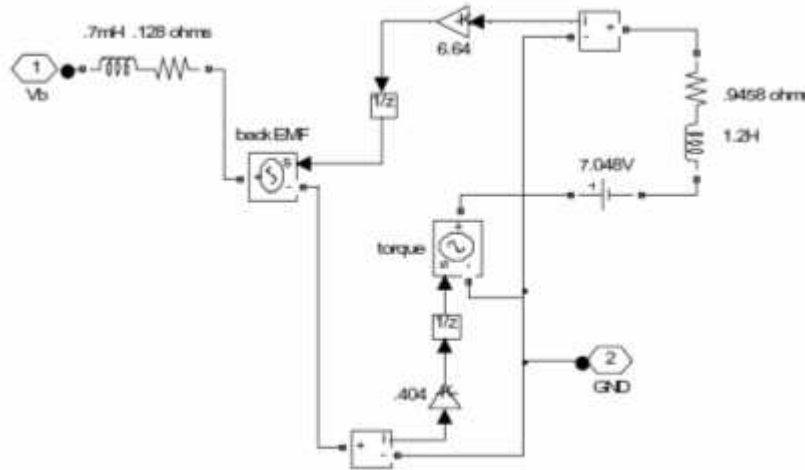


**Figure 3.36: Cisco Wireless Router [35]**



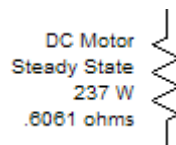
### 3.2.8 New DC Motor Model

Many prior students had designed very accurate and complex models of the DC motor similar to the model in Figure 3.37 :



**Figure 3.37: Simulink Model of a DC Motor [36]**

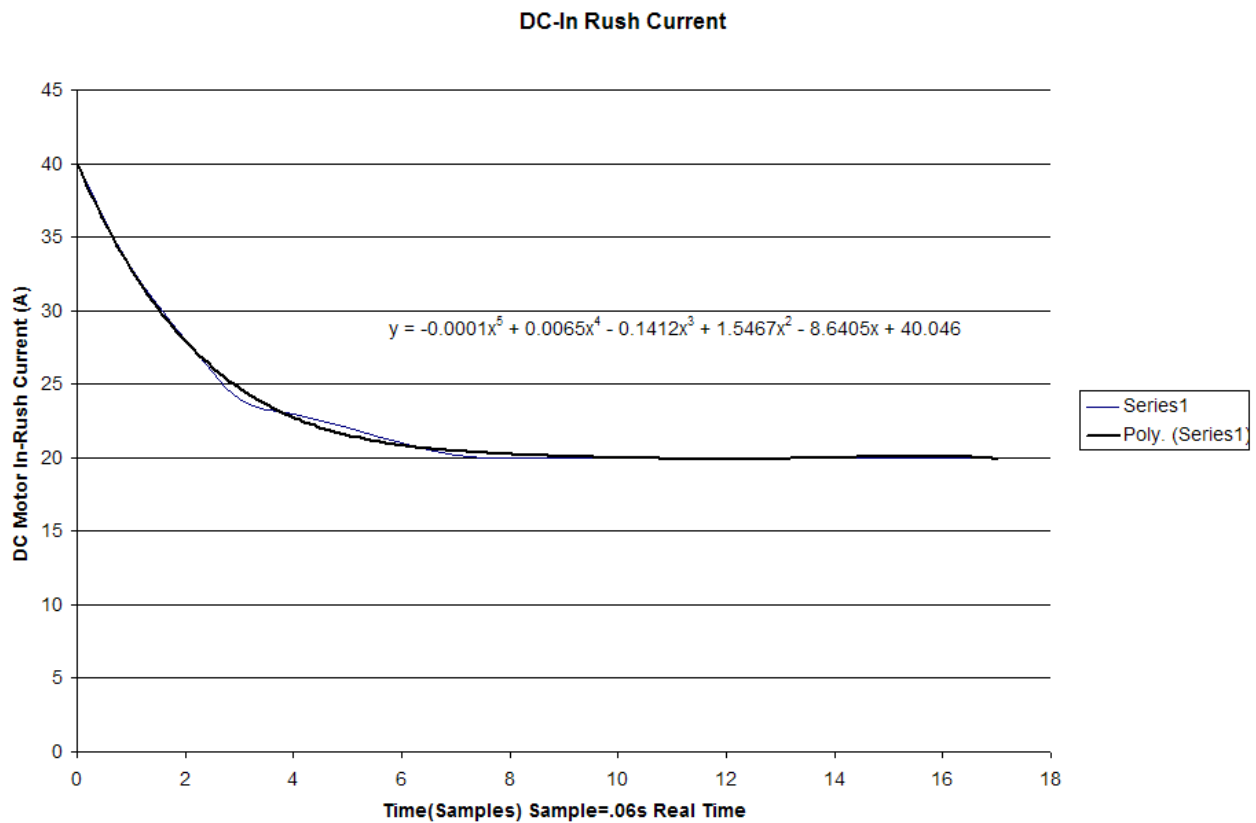
However, in most, if not all cases when the entire simulation needed to be run, it was discovered that having more differential elements led to increased simulation times. The degradation in simulation time was severe enough that Sheffield [1] decided to use a simple resistor to model the DC motor. Another justification in his use of this DC motor model was the assertion that the ultra capacitor would cancel out the transient effects of the DC motor. Figure 3.37 shows the model that was used in the previous simulation.



**Figure 3.38: Previous DC Motor Model [1]**

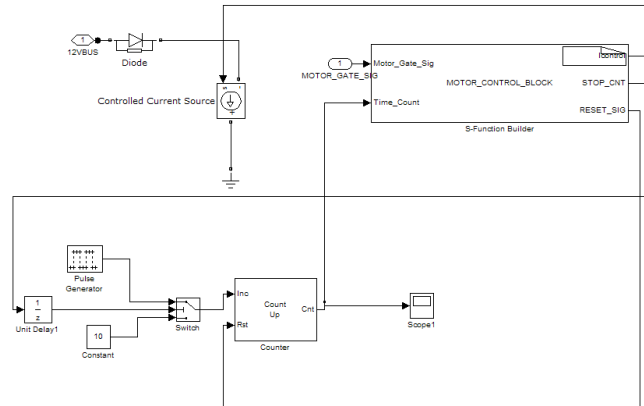
Choosing a resistor meant that the simulation time would not be an issue; however, choosing a resistor does not account for starting currents associated with starting a DC motor.

The decision was made to model the DC motor by using experimental data as taken by Witts [36]. Using purely mathematical formulas rather than differential elements would allow the transient effects to be modeled while incurring none of the simulation penalties. In Figure 3.38 Witts [36] recorded data for the in-rush current for a given mechanical load of 8in-lbs for 1000ms. Also, in Figure 3.39 is the curve of best fit for the actual data.



**Figure 3.39: DC Motor In-rush Current**

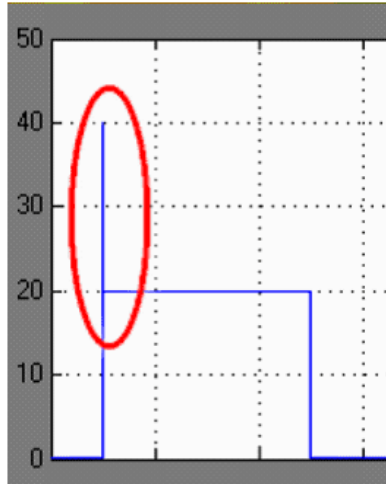
In order to implement this function in Simulink an S-function block along with a controllable counter (see Figure 3.40) was used. The code for this is shown in Appendix A.7 Motor Model Control Block S-Function Block Code.



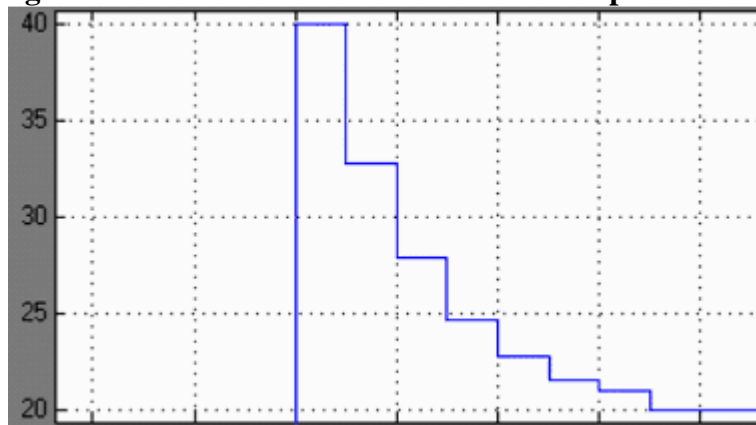
**Figure 3.40: Simulink Implementation of DC Motor**

The motor control block essentially starts or stops the count based upon the motor gate signal. The output of the motor control block is a current controlled source that acts like a variable resistor. The current control portion essentially allows the model to represent the electrical transient effects of the DC motor.

As seen in Figure 3.41 and Figure 3.41 the model accurately reflects the transient starting effects of the DC motor. In Figures 3.40 and 3.41 the y-axis is the motor current draw in amps and the x-axis is time.



**Figure 3.41: DC Motor Model Transient Output Current**



**Figure 3.42: DC Motor Model Transient Output Current Zoom In**

The accuracy of the model was limited primarily to the step size of the simulation. The smaller step size would give a finer resolution, but also it would have decreased the simulation time. A step time of .001 (60ms real time) was used to accomplish the goal of modeling the transient effects of the DC motor while not compromising the simulation speed significantly.

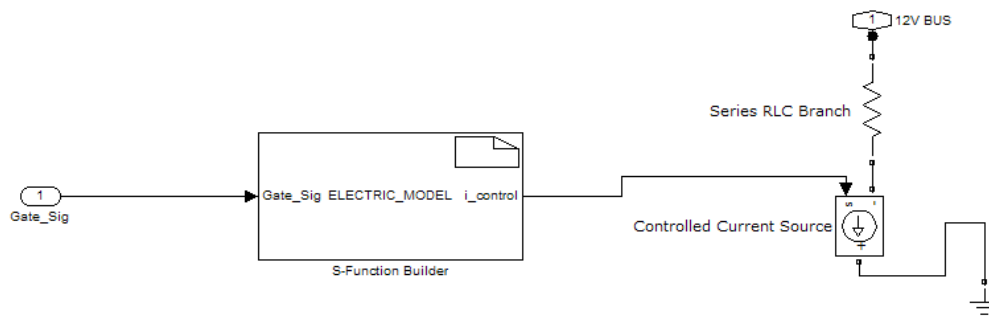
### 3.2.9 New Thermoelectric Model

The new thermoelectric model was split into two components. The electric model interfaces to the 12V bus and approximates the power consumed by the thermoelectric

cooler. The second portion is the thermal model, which reflected the internal temperature of the cooler based upon experimental data.

## Electric Model

In the electric model it is assumed that the T.E. cooler will consume the same amount of power regardless of the outside air temperature or the temperature inside the cooler. While, this assumption is not necessarily true, the experimental data showed that the power consumption varied only from 60W to 70W over the temperature range measured [1]. Therefore keeping this in mind, the model in Figure 3.43 was created:



**Figure 3.43: Electric Model of T.E. Cooler**

The current consumed by the load is controlled via an S-function block which can be adapted in the future to include a more thermodynamically correct model. The code for this block is shown in Appendix A.9 T.E. Cooler Electric Model S-Function Code.

## Thermal Model

The thermal model was a bit more challenging to actually implement. There were in fact multiple situations to consider (see Table 3.4).

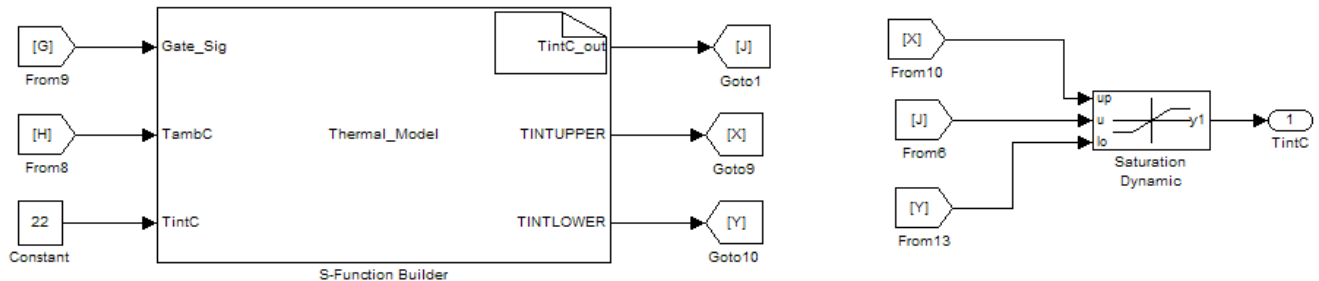
**Table 3.4: Thermal Model Table**

CASE 1	T.E. COOLER ON	$T_{amb} < T_{int}$
CASE 2	T.E. COOLER ON	$T_{amb} > T_{int}$
CASE 3	T.E. COOLER OFF	$T_{amb} < T_{int}$
CASE 4	T.E. COOLER OFF	$T_{amb} > T_{int}$

**T<sub>in</sub>**-Internal temperature of the cooler (°C)

**T<sub>amb</sub>**- Ambient temperature of surroundings (° C)

Sheffield [1] had found the heating and cooling rates for cases 2 and 4. However, cases 1 and 3 were not specifically measured. Therefore, it was assumed that the heat lost in case 1 would be equal to the heat gained in case 4. This heat loss would aid the cooler and the temperature loss would be more rapid. In case 3, it was assumed that the heat lost from the cooler would be the same as in case 1 with the exception that the T.E. cooler was not removing heat.



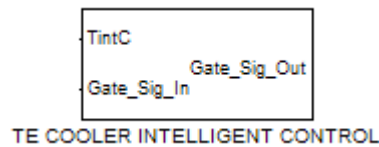
**Figure 3.44: T.E. Cooler Thermal Model**

The thermal model was implemented using an S-function control block (Code shown in Appendix A.8 T.E. Cooler Thermal Model S-Function Code) along with a dynamic saturation limit on the internal cooler temperature. The saturation block was necessary since according to the data sheets the T.E. cooler could only cool down to

about 25°C below the ambient temperature. The saturation block placed a finite limit on the internal temperature, which would be reflective of what would actually occur.

### **Thermoelectric Cooler Intelligent Controller**

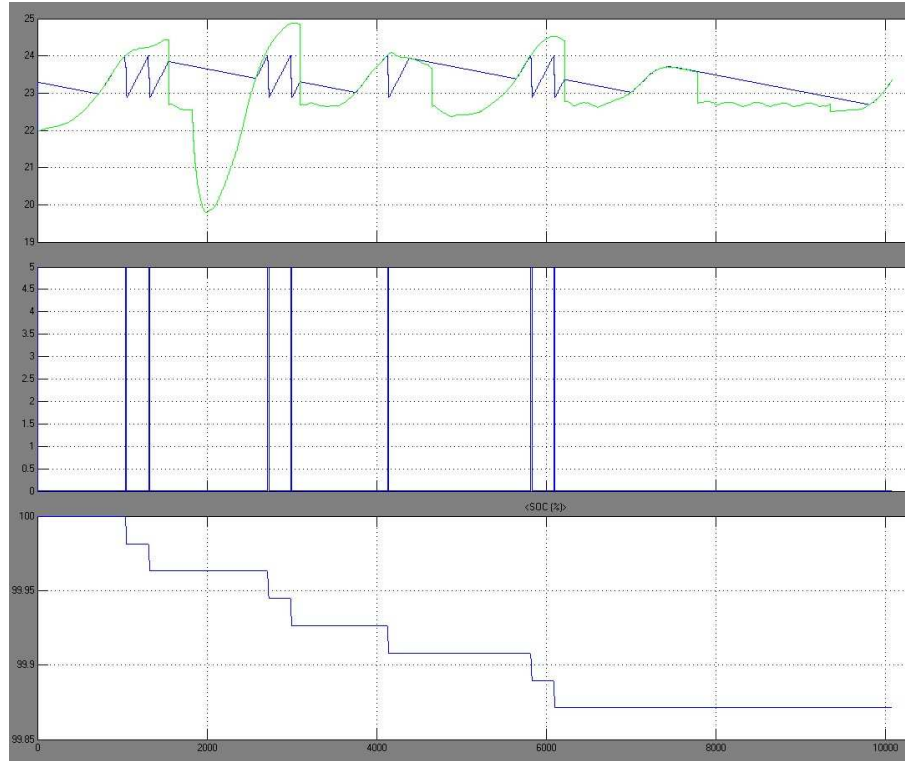
Sheffield [1] mentioned that a control scheme could be developed where the T.E. cooler could be turned on and off based on temperature set points. The idea would be to minimize the power consumed by the T.E. cooler. Research on battery modeling indicated that for lead-acid batteries if the battery temperature is maintained at about 25°C then the life of the battery can be significantly extended. In fact it was found that battery life can be extended by as much as double the life of a battery that is not temperature controlled [3]. Taking this into account a smart temperature controller was created (see Figure 3.45) that would essentially regulate the T.E. cooler's internal temperature within a specified range.



**Figure 3.45: T.E. Smart Controller Block Diagram**

### **Thermoelectric Cooler Model Output**

To evaluate the overall performance of the model ambient temperature data over the seven day period of interest was used. The results are shown in Figure 3.46.



**Figure 3.46: T.E. Temperature Control Results**

On the top chart in green is the ambient temperature and in blue is the cooler's internal temperature. The middle graph is the current required by the T.E. cooler. The graph at the bottom is the SOC of the battery. Looking at the first portion, when the 24°C set point is reached, the cooler turns on which drives down the temperature. When the lower set point is reached the cooler turns off and the temperature is allowed to rise again. The primary advantage of the control scheme is proven because there is reduced power consumption, and thus the overall system has more power to service other loads.

### 3.2.10 Switch Model and Intelligent Load Control

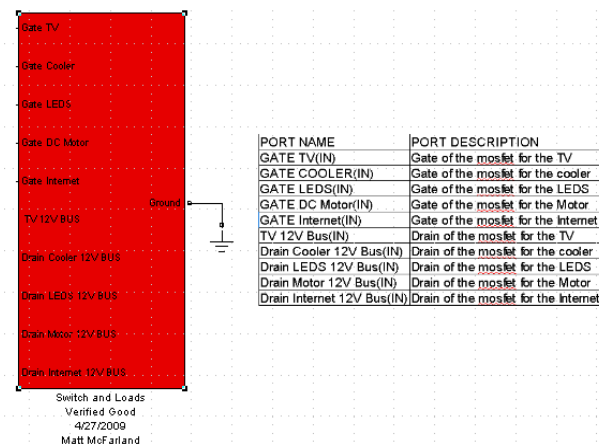
The load model of the system was initially based strictly on power consumption and the anticipated loading times. A total of 6 scenario types were written as seen in



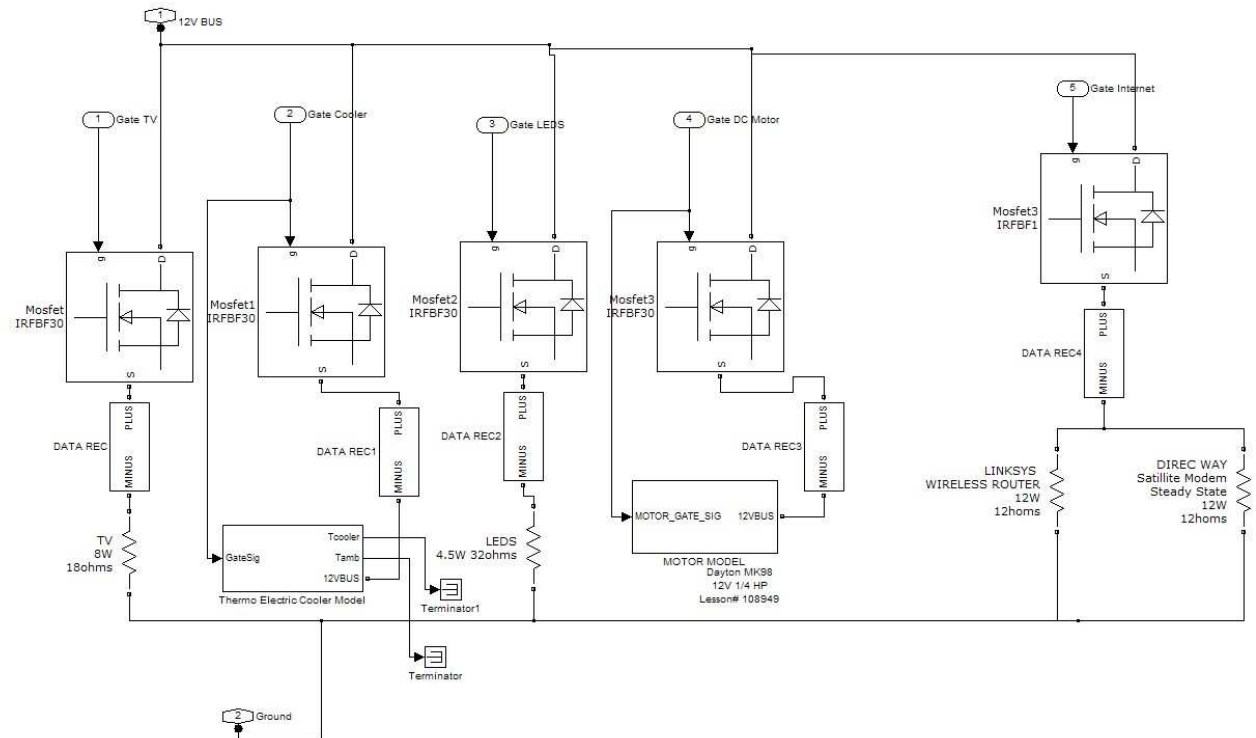
Chapter 4. The loads and switches are broken up into one block while the load control is placed in another block.

## Switch Model

The switch and load block model was based directly on the existing hardware. In other words, the MOSFET switches used in this model were given identical characteristics to the data sheet of the IRBF30. Shown in Figure 3.47 is a diagram of the overall model of this block. Like many of the models built, this model uses subsystems to allow for viewing the design in layers of abstraction, simplify debugging, and to increase modularity. Now going further into this block (as shown in Figure 3.48) it can be seen how each circuit element is broken up into its equivalent circuit.



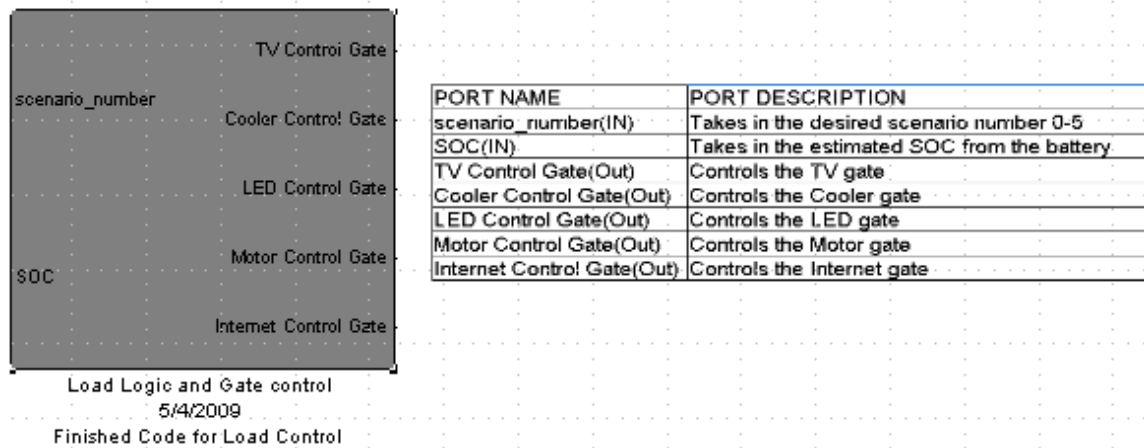
**Figure 3.47: Switch and Loads Block Diagram**



**Figure 3.48: Switch and Loads Detail**

## Load Control Model with Intelligent Load Control

The load control module was built to drive the gate signals of the MOSFET switches. The load control also drives the appropriate loads at the correct time, and as a helpful feature it now has the ability to override the load schedule if the estimated SOC of the battery drops below 80%. Cutting off all non-essential loads will allow the battery life to be preserved. Shown in Figure 3.49 is the overall block diagram of the load control block.



**Figure 3.49: Load Logic and Gate Control Block Diagram**

The load control block is controlled primarily by an S-function block written in C-code as shown in Appendix A.10 Scenario Load Control S-Function Block Code.

## **Chapter 4: Simulation**

### **4.0 Chapter Overview**

This chapter includes some notes on setting up the simulation environment, along with assumptions for running the simulation over a seven day scenario. The results for the seven day simulation and an evaluation of the simulation results are also presented.

### **4.1 Setting up the Simulation Environment**

#### **Compiler Setup**

Usually if the simulation is being setup on a new machine the compiler must be setup first (used in S-functions). A step by step guide to setup the compiler is shown in Appendix E.

#### **S-Function Setup**

In most, if not all cases, the S-functions will need to be rebuilt (on a computer that has never run this simulation) in order to function correctly. To rebuild the S-functions see the example in Appendix E. There are quite a few S-function blocks; therefore it is important to make sure all blocks are rebuilt.

#### **Simulation Controls**

In order to facilitate ease of use, most of the controls and data recording items are located in a common location of the simulation (see Figure 3.1).

Additional data recording items are included inside some of the various blocks. Listed in Table 4.1 are all the various pieces of output data that are recorded every time a simulation is run.

**Table 4.1: Data Recording Items**

FILE NAME	DESCRIPTION
vbatt.mat	Battery voltage(V)
soc_7days.mat	SOC of the battery (%)
batt_curr.mat	The current into the battery(A)
MOTOR_LOAD.mat	The current into the motor(A)
LED_LOAD.mat	The current into the LEDs(A)
INTER_LOAD.mat	The current into the internet loads (A)
COOLER_LOAD.mat	The current into the thermoelectric cooler (A)
HUT_RADIO.mat	The current into the radio from the huts (A)
HUT_LEDS.mat	The current into the LEDs from the huts (A)
HUT_CELL_PHONE.mat	The current into the cell phone chargers from the huts (A)

The controls items are as follows:

- 1) Selection between PV/WIND or PV and WIND
- 2) HVDC Link (Microgrid) ON/OFF control
- 3) Ultra-capacitor ON/OFF control
- 4) Spartan 3E or Laptop control source selector
- 5) Scenario Number selection.
- 6) Intelligent Load Control ON/OFF

## 4.2 Sample Times and Data Recording

### Real time, Sample Time, and Simulation time

When deciding to run this simulation it is important to understand the relationship between real time and sample time. Currently, the sample time relates to the real time based upon the input insolation data which is updated every minute. This sample is equal to 1 minute of real time. Now, simulation time (how long it will take you to run a given simulation) is dependent upon the overall simulation sample time.

Currently the simulation sample time is set to .001 (60ms real time), which is a small enough time period so that the starting current from the DC motor can be calculated by the system. As the sample time is made smaller, the length of the simulation time will

go up. In other words an overall system sample time of .1 will complete simulating much faster than an overall system sample time of .001. Ideally, the system sample time should be as small as possible for accuracy; however the primary thing that dictates the simulation sample time is the smallest time period you need to represent.

To find the step size needed to model the transient effects of the old DC-DC converter model an update rate of at least 2 times the pulse width modulation frequency would be needed due to the Nyquist criterion. In this case the pulse width modulation frequency of the universal DC-DC converter is 30 KHz [32]. Shown in (17) is the required calculation.

$$f_{sample} = 2 f_{pwm} = 60Khz \quad (17)$$

$$\frac{1 \text{ sample (simulation time)}}{1 \text{ minute (real time)}} = \frac{X \text{ (sample time)}}{\left( \frac{1}{60Khz} \right) \left( \frac{1 \text{ minute}}{60 \text{ s}} \right)}$$

$X = 277.77E-9 \text{ (Simulation Step Size required)}$

### **Data Recording and Limits of Simulink**

While it is desirable to have as much data as possible, Simulink actually has physical limits for how much data it can actually record. Shown in the Table 4.2 (from Simulink's help file), is a list of limits.

**Table 4.2: Simulink Limits Table**

Model Feature	Limit
Maximum number of levels in a block diagram	1024
Maximum number of branches in a line	1024
Maximum length of a parameter name	256
Maximum length of a parameter string value	32768
Maximum value of a model window coordinate	32768
Maximum number of bytes of logged simulation data	2 <sup>31</sup> -1 bytes on 32-bit systems, 2 <sup>48</sup> -1 bytes on 64-bit systems
Maximum length of integer and fixed-point data types	128 bits

Currently, all items that record data are set to record data every sample, which is 1 minute of real time. If the sample time of the data recorders is lowered then samples will be taken more frequently and then more memory space will be taken up. If data is taken at a rate equal to the system step size (.001 or 60ms of real time), then it was found that when the full 7 day simulation is run, then the simulation crashes due to memory limitations.

### **4.3 Requirements for One Week Scenarios.**

#### **4.3.1 Possible Load Schedules**

Listed in Figure 4.1 through Figure 4.4 are the possible load schedules available to the system. However, modifications can be made to consider many more cases.

TYPE1	TV	Cooler	LED	DC Motor	Internet	TYPE2	TV	Cooler	LED	DC Motor	Internet
	0		X				0				
	100		X				100				
	200		X				200				
	300		X				300				
	400		X				400				
	500		X				500				
	600		X				600				
	700		X				700				
PV on	800	X	X		X	PV on	800				
	900	X	X		X		900				
	1000	X	X		X		1000				
	1100	X	X		X		1100				
	1200	X	X	X	X		1200			X	
	1300	X	X		X		1300			X	
	1400	X	X		X		1400				
	1500	X	X		X		1500				
	1600	X	X		X		1600				
	1700	X	X		X		1700				
PV off	1800		X	X		PV off	1800			X	
	1900		X	X			1900			X	
	2000		X	X			2000			X	
	2100		X	X			2100			X	
	2200		X				2200				
	2300		X				2300				

**Figure 4.1: Primary Load Schedule Scenarios 1 and 2**

TYPE3	TV	Cooler	LED	DC Motor	Internet	TYPE4	TV	Cooler	LED	DC Motor	Internet
	0		X				0	X	X		
	100		X				100	X	X		
	200		X				200	X	X		
	300		X				300	X	X		
	400		x				400	x	X		
	500		X				500	X	X		
	600		X				600	X			
	700		X				700	X			
PV on	800	X				PV on	800	X			
	900	X					900	X			
	1000	X					1000	X			
	1100	X					1100	X			
	1200	X					1200	X			
	1300	X					1300	X			
	1400	X					1400	X			
	1500	X					1500	X			
	1600	X					1600	X			
	1700	X					1700	X			
PV off	1800	X	X			PV off	1800	X	X		
	1900	X	X				1900	X	X		
	2000	X	X				2000	X	X		
	2100	X	X				2100	X	X		
	2200	X					2200	X	X		
	2300	X					2300	X	X		

**Figure 4.2: Primary Load Schedule Scenarios 3 and 4**

TYPE5	TV	Cooler	LED	DC Motor	Internet	TYPE6	TV	Cooler	LED	DC Motor	Internet
	0		X		X		0				
	100		X		X		100				
	200		X		X		200				
	300		X		X		300				
	400		X		x		400				
	500		X		X		500				
	600				X		600				
	700				X		700				
PV on	800				X	PV on	800				
	900				X		900				
	1000				X		1000				
	1100				X		1100				
	1200				X		1200				
	1300				X		1300				
	1400				X		1400				
	1500				X		1500				
	1600				X		1600				
	1700				X		1700				
PV off	1800		X		X	PV off	1800	X		X	
	1900		X		X		1900	X		X	
	2000		X		X		2000	X		X	
	2100		X		X		2100	X		X	
	2200		X		X		2200	X		X	
	2300		X		X		2300	X		X	

**Figure 4.3: Primary Load Schedule Scenarios 5 and 6**

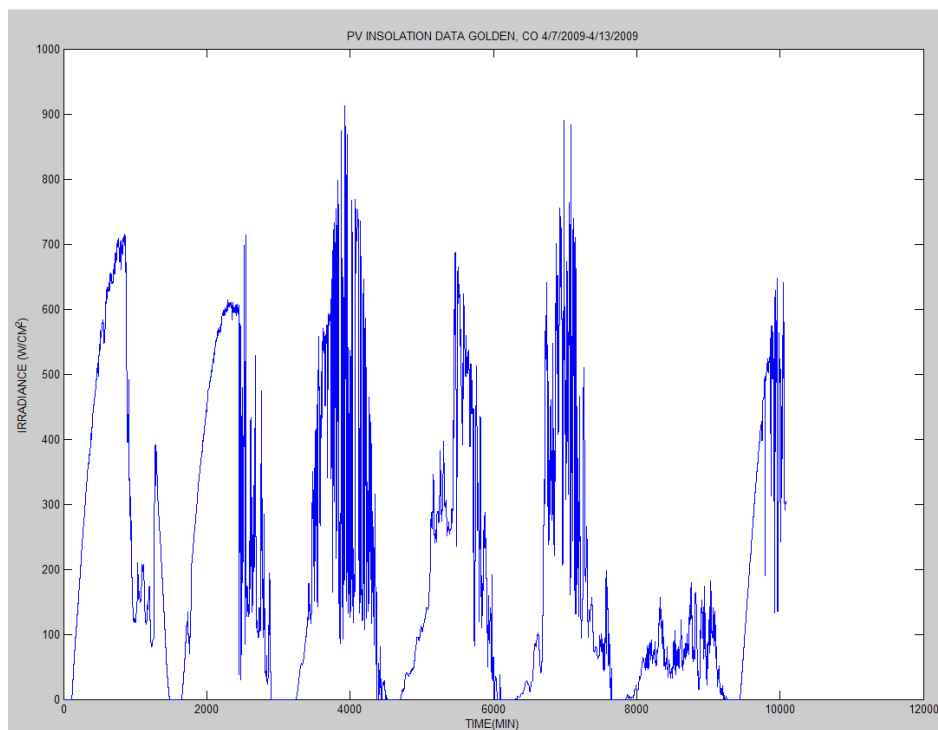


		RADIO	CELL	LEDS
	0			
	100			
	200			
	300			
	400			
	500			
	600			
	700			
PV on	800			
	900	X		
	1000	X		
	1100	X		
	1200	X		
	1300	X	X	
	1400	X	X	
	1500	X		
	1600			
PV off	1700			
	1800			
	1900			
	2000			X
	2100			X
	2200			X
	2300			

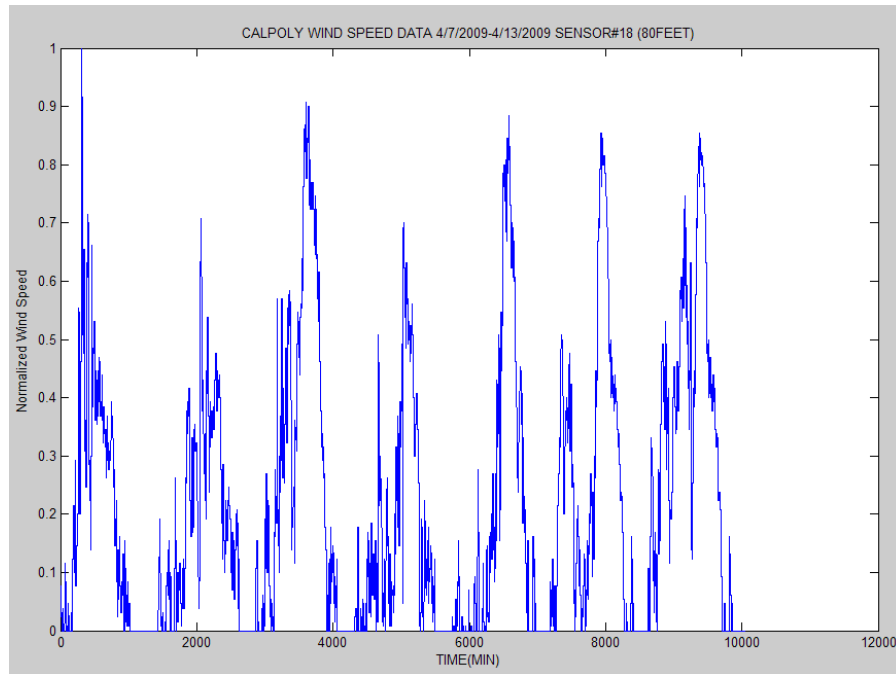
**Figure 4.4: Microgrid Load Schedule**

### 4.3.2 Input Data

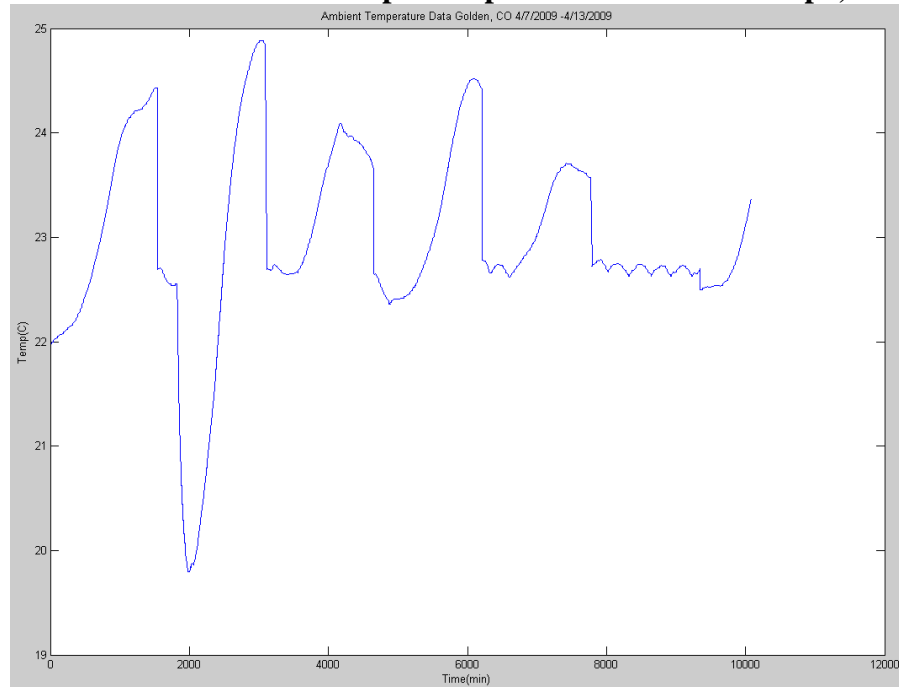
The input data chosen was selected over identical date ranges (4/7/2009-4/13/2009) to ensure the most realistic case for that date range. The primary pieces of input data were insolation, wind speed, and ambient temperature as shown in Figure 4.5, Figure 4.6, and Figure 4.7.



**Figure 4.5: PV Insolation Data Golden, CO [7]**



**Figure 4.6: Normalized Wind Speed Input Data San Luis Obispo, CA [30]**



**Figure 4.7: Ambient Temperature Data Golden, CO [7]**

### **4.3.3 Assumptions**

1. Initial Battery Conditions: SOC 100%  $V_{batt}=13.9V$
2. Initial Ultra-capacitor Conditions: Fully Charged  $V_{cap}=13.9V$
3. Scenario 1 only is being run.
4. The Spartan 3E will be used as control; therefore the Spartan 3E load will be used.
5. Step Size will be .001 (60ms of real time). Input data is updated every 1 min of real time.
6. The Standalone simulation will simply include the primary loads (LEDs, DC Motor, T.E. Cooler, TV, and Internet) and no microgrid load on the system.
7. The microgrid simulation will include the primary loads on the system as well as the microgrid loads.
8. Both the intelligent load control (ILC) and non-intelligent load control (NILC) will be evaluated for each case.

### **4.3.4 Load Scenario Number**

After running all the possible scenario numbers for each of the cases shown in this report (72 cases), it was found that the worst case (drained the battery the most) was scenario number 4. This is due primarily to the constant loading of the T.E. cooler. However, after completion of the new T.E. temperature controller module the constant load went away. Instead scenario number 1 became the worst case due to the motor running once every day. Therefore it was decided to run scenario 1.

## **4.4 Scenarios**

### **4.4.1 PV Standalone**

The PV standalone simulation will have the PV driving the system using seven days of isolation data as shown in Figure 4.5.

### **4.4.2 PV and Microgrid**

This simulation will evaluate if the PV can successfully supply power to all of the existing loads and the microgrid.

### **4.4.3 Wind Turbine Standalone**

In this simulation the wind turbine will be the only driving power source, the data used for this is shown in Figure 4.6.

### **4.4.4 Wind Turbine and Microgrid**

This simulation will evaluate if the wind turbine given the input wind data can supply power to the existing loads and the microgrid.

### **4.4.5 PV and Wind Turbine Standalone**

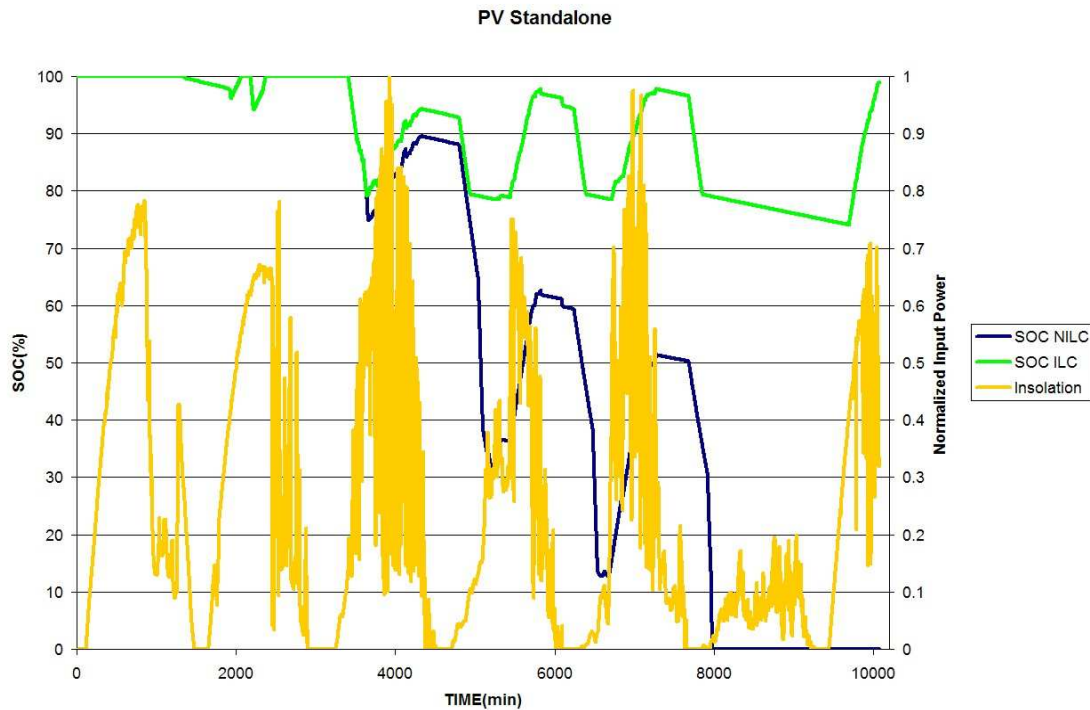
In this simulation the PV and the wind turbine will be run. As per the wind turbine user manual it is recommended that the wind turbine never be run while another DC-DC converter is attempting to regulate the system voltage [28]. Therefore a rudimentary switching block was created that switched to the wind turbine whenever the PV insolation data dropped to zero (night time). This device ensured that only one power source would drive the system at any one time.

#### 4.4.6 PV, Wind Turbine, and Microgrid

This simulation will evaluate if the combination of the wind turbine and the PV using the switching algorithm can successfully supply power to the existing loads and the microgrid.

### 4.5 Results

#### 4.5.1 PV Standalone



**Figure 4.8: PV Stand Alone SOC Results**

### 4.5.2 PV and Microgrid

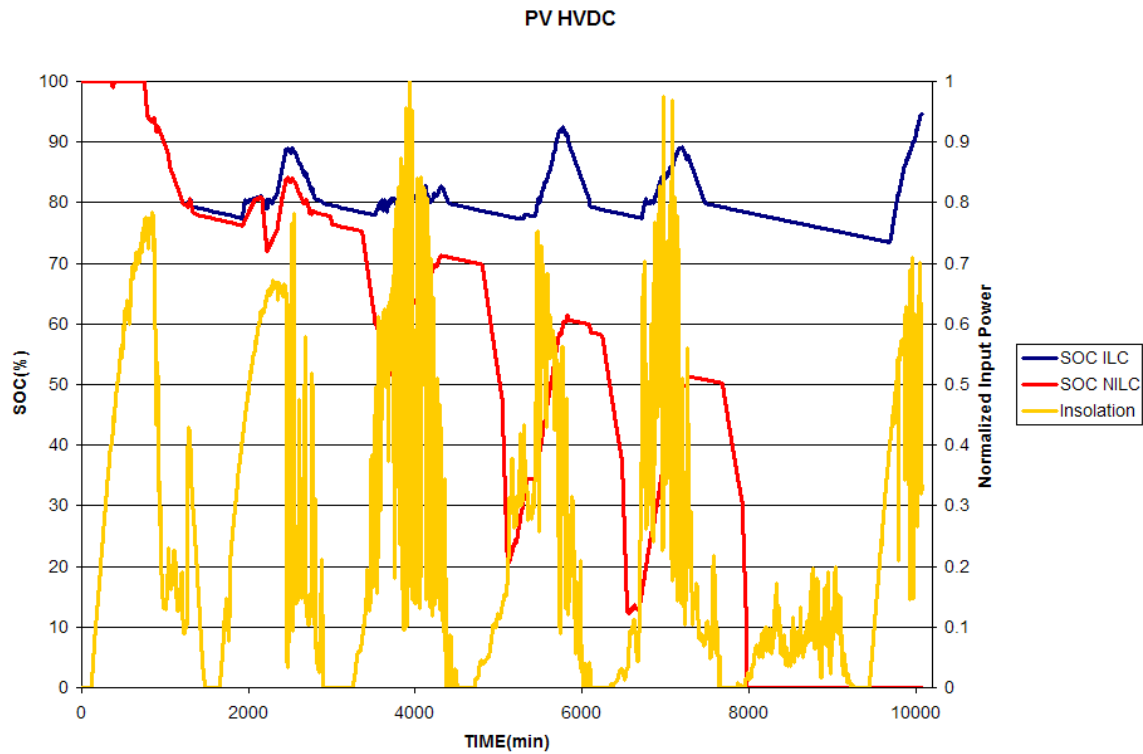


Figure 4.9: PV Microgrid SOC Results

### 4.5.3 Wind Turbine Standalone

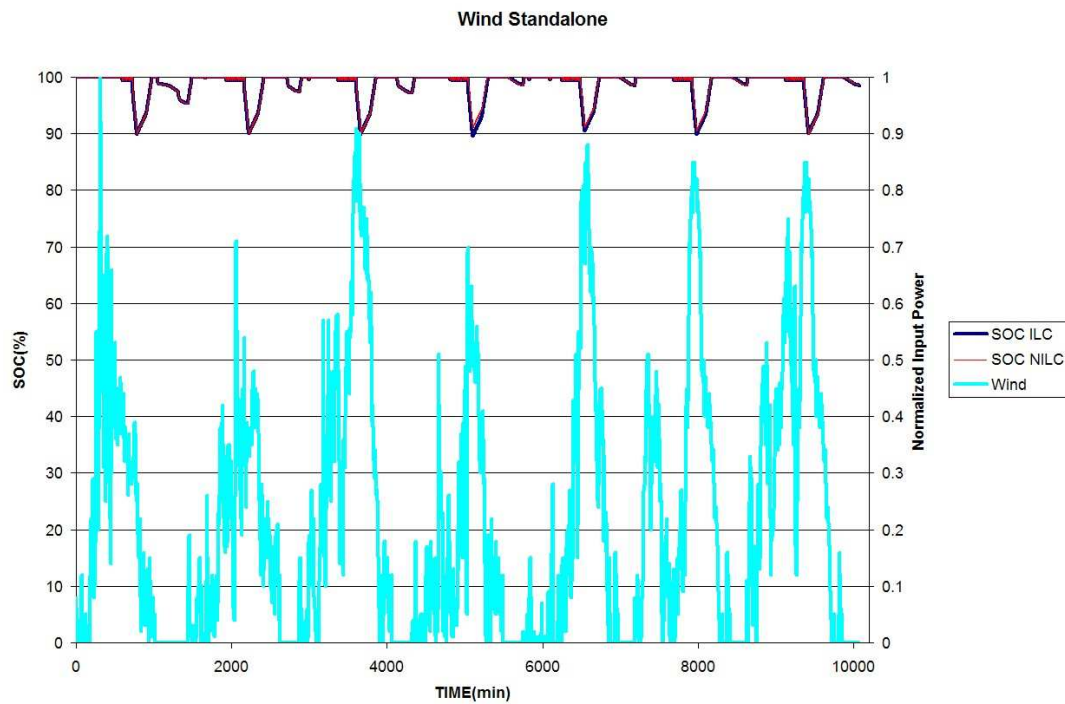


Figure 4.10: Wind Turbine Standalone SOC Results

#### 4.5.4 Wind Turbine and Microgrid

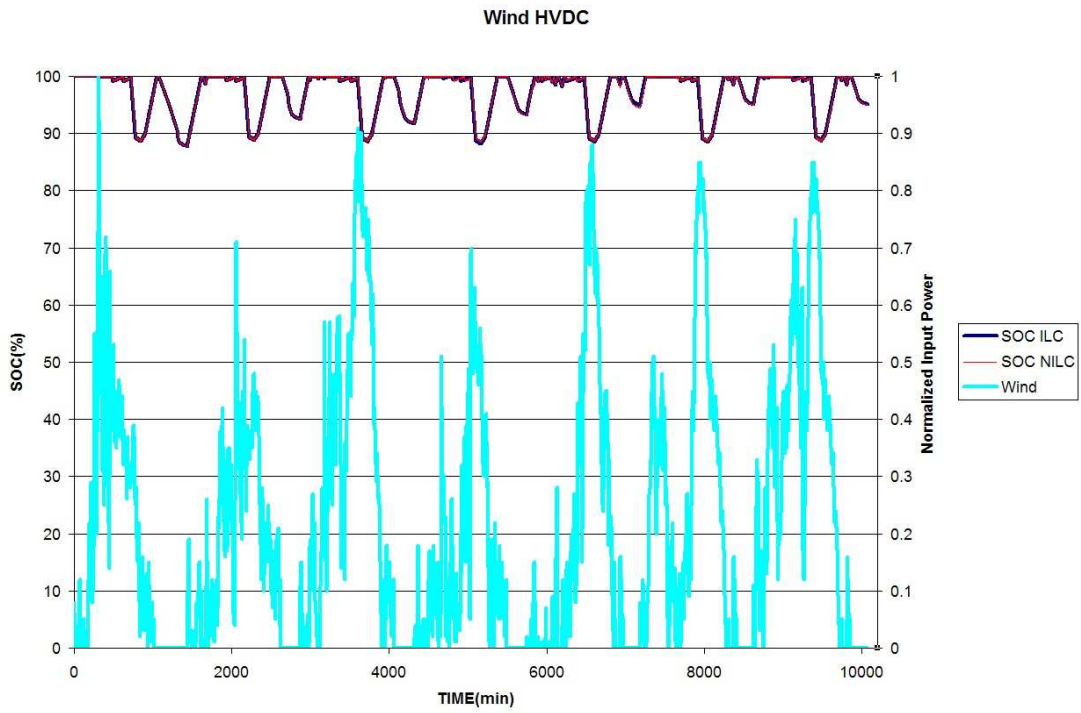


Figure 4.11: Wind Turbine and Microgrid SOC Results

#### 4.5.5 PV and Wind Turbine Standalone

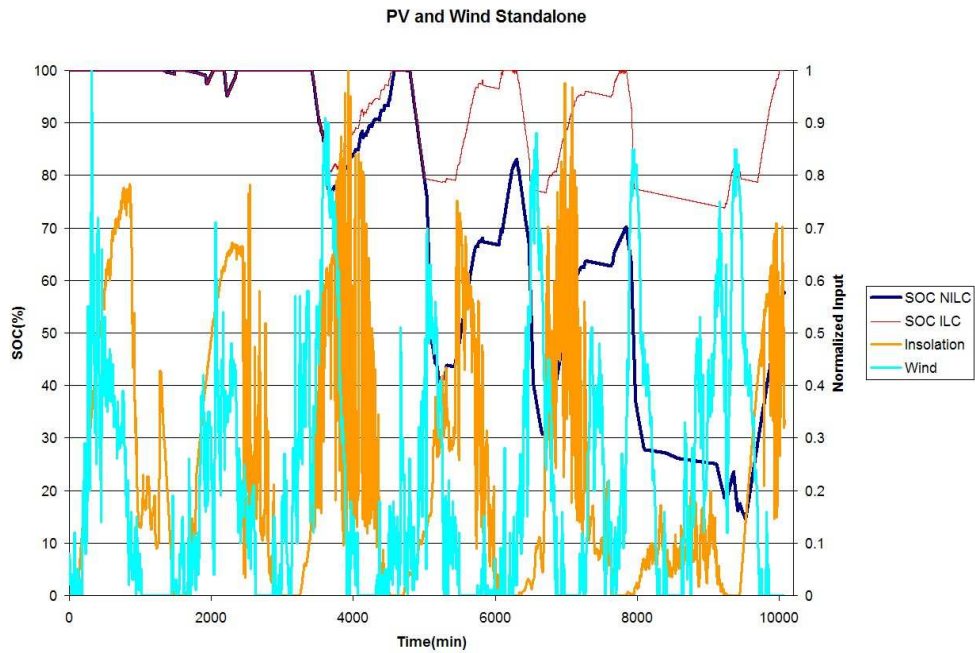


Figure 4.12: PV and Wind Turbine SOC Results

#### 4.5.6 PV, Wind Turbine, and Microgrid

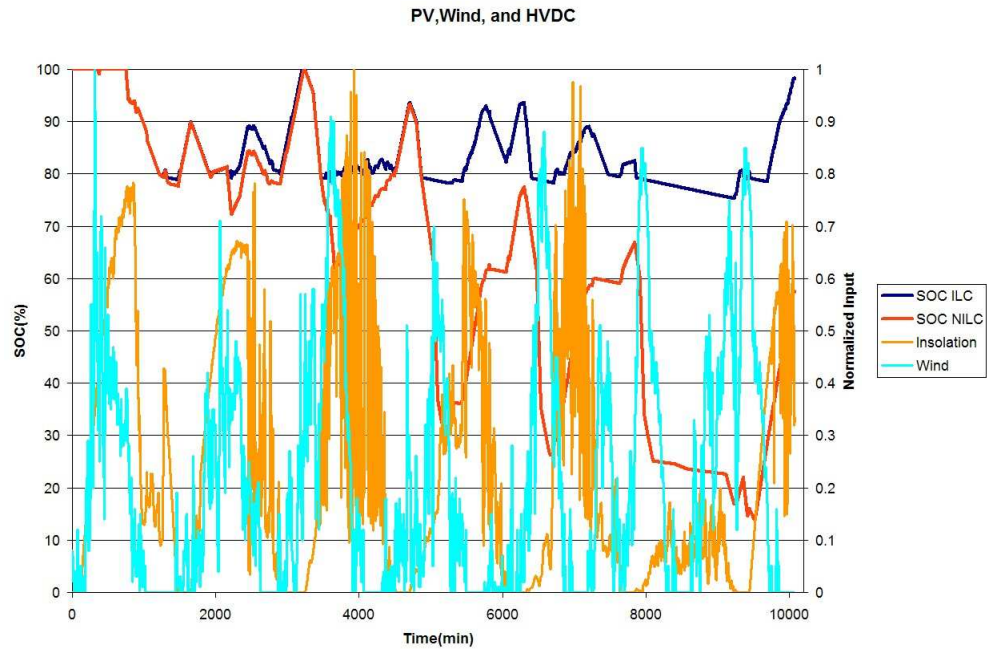


Figure 4.13: PV, Wind, and Microgrid SOC results

#### 4.5.7 Summary of Results

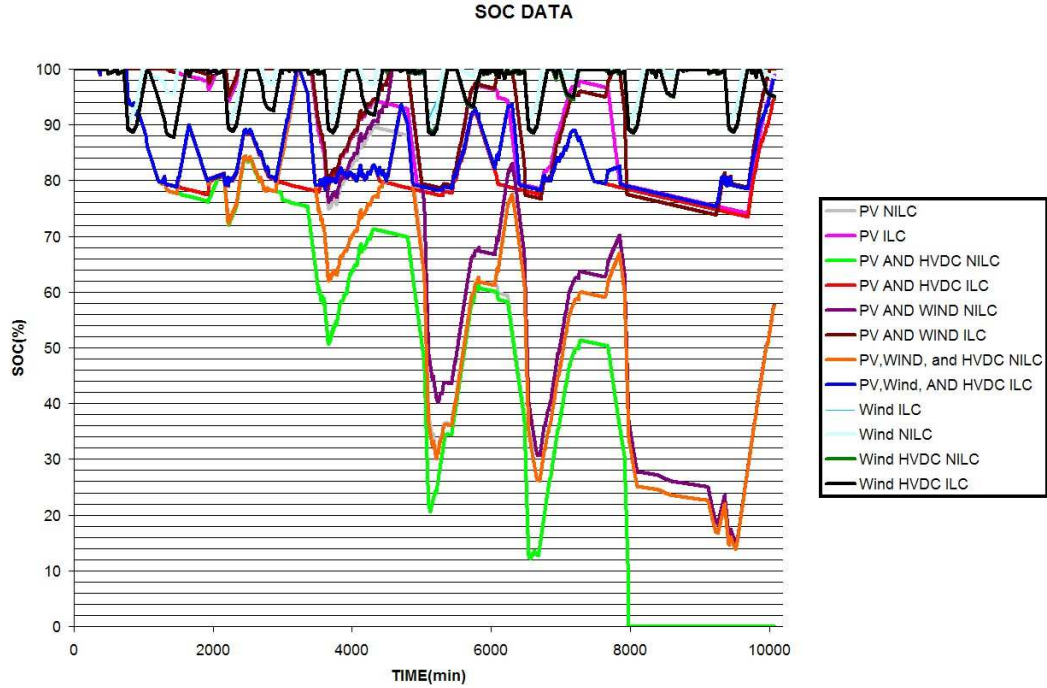
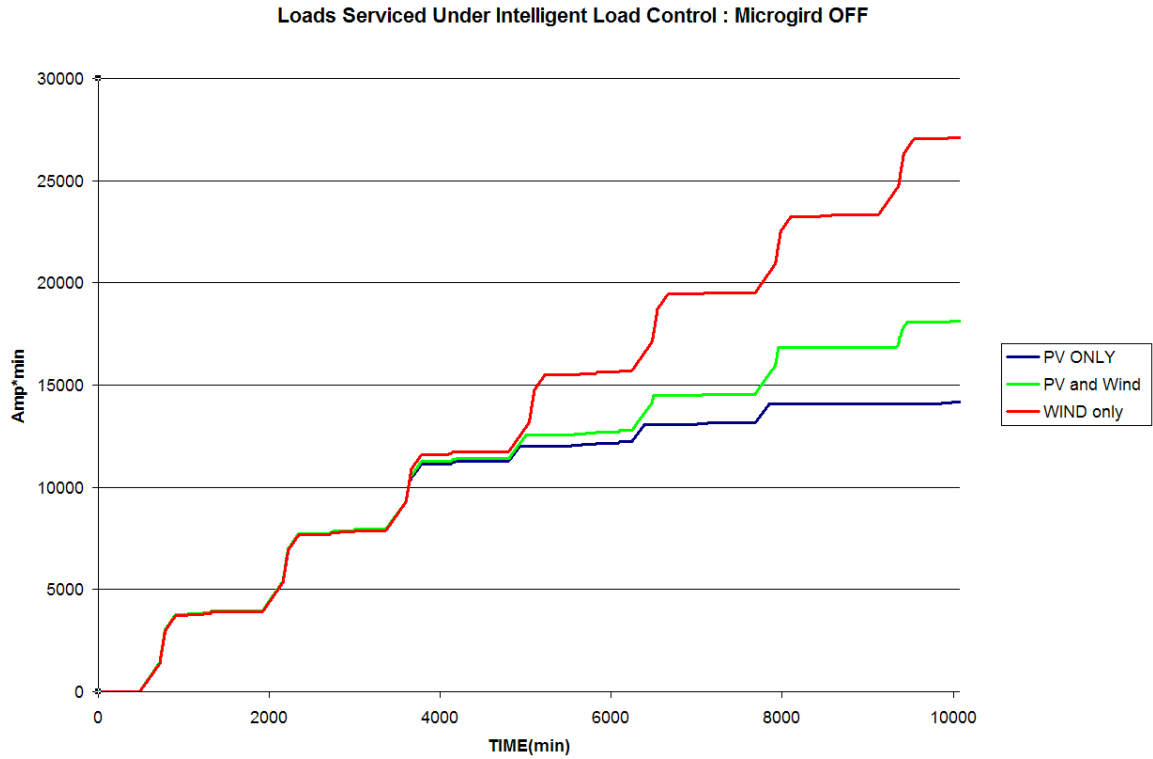
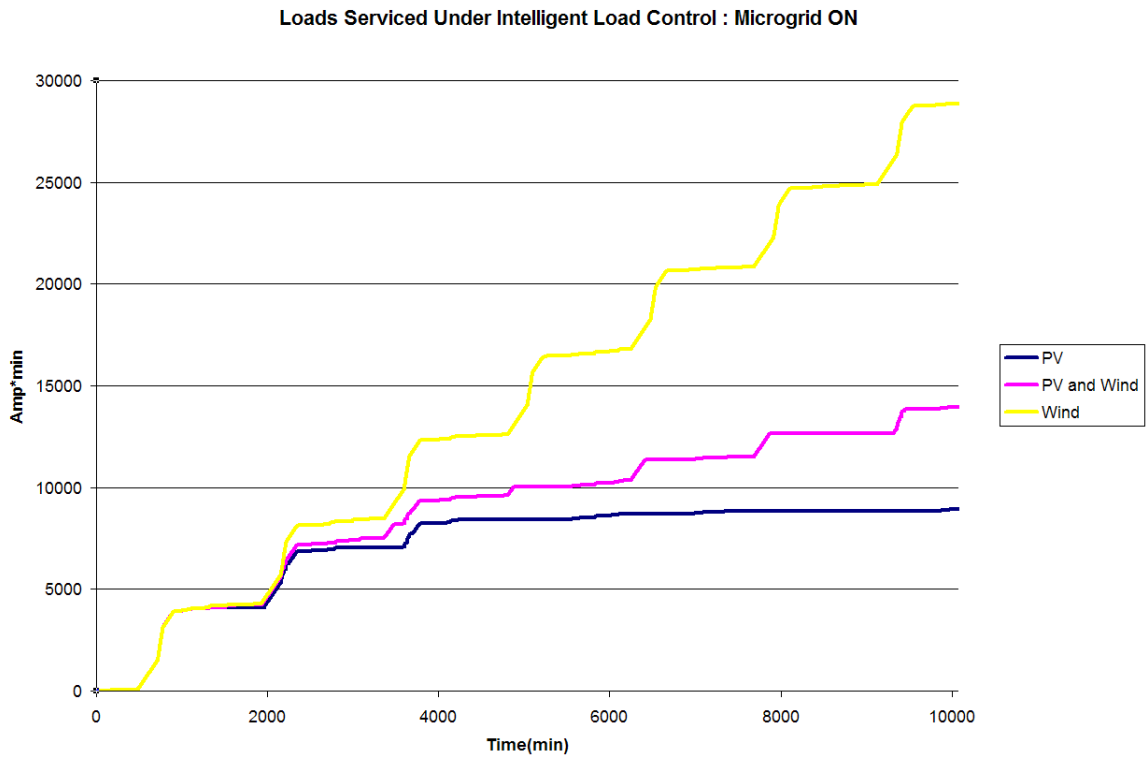


Figure 4.14: SOC Summary Chart





**Figure 4.15: Loads Serviced Chart Microgrid Off**



**Figure 4.16: Loads Serviced Chart Microgrid On**

## **4.6 Evaluation:**

### **4.6.1 PV Standalone**

In this case the results are rather dramatic. The non-intelligent load control system is able to maintain the battery charge sufficiently up until about the beginning of the fourth day. Coincidentally, the PV insolation drops off significantly on the fourth day. During the first two days the insolation increased rather rapidly; however, on the fourth day the insolation lags and therefore the SOC of the battery suffers. This coupled with the fact that the primary loads turn on around the same time (including the motor), this situation quickly becomes a bad situation for the battery. Finally, in the end the battery SOC goes to zero. In reality this situation would lead to excessive sulfate buildup on the electrodes and the battery life would quickly be shortened.

However, when looking at the intelligent load control case, the battery life is preserved and ends up at a SOC of almost 100%. The intelligent controller is completely shutting off all loads (with the exception of the control items) when the SOC drops below 80%. This control protects the battery from excessive discharge and extends the battery life. The tradeoff however, is that not all the loads are getting serviced and therefore the system does not do as much.

### **4.6.2 PV and Microgrid**

In this case the PV is powering all of its normal loads in addition to the microgrid, which is approximately 60W of power consumption. However, the microgrid has a built-in intelligent load control which shuts itself down if the SOC drops below 80% regardless of the state of the intelligent load control selector.

As anticipated, the SOC drops off rather rapidly above 80%, then the DC microgrid shuts down and only the primary loads are consuming power at that point. Similar to the PV standalone the SOC ends up at 0% due to the poor insolation performance and the excessive loads being placed on the system. When the intelligent load control is utilized the battery SOC once again is maintained; however, it appears that there are only short periods of time where the system can actually service the loads placed upon it.

#### **4.6.3 Wind Turbine Standalone**

For the wind turbine case it appears that due to the input wind data, the wind turbine performed extremely well at keeping the battery charged and servicing the loads. The ILC and NILC are identical because the wind turbine never lets the SOC drop below 80%. When the wind turbine was sized originally the assumption was that the wind turbine power output would only be a fraction of what its maximum could actually output. During this particular data set the wind turbine was putting out approximately 70-80% of its maximum power on average, which is 480W. That is four times more power output than the PV at its maximum solar insolation (which only occurred for short periods).

#### **4.6.4 Wind Turbine and Microgrid**

Comparing this case to the wind turbine standalone case, it can be seen that while the SOC drops more as anticipated, the SOC never drops below 80%. The most the SOC drops to is 86%. Again, the intelligent load control does not come into play due to the

80% set point. In this case the wind turbine is able to supply all the load power requested of it plus it maintains the battery at an adequate charge level.

#### **4.6.5 PV and Wind Turbine Standalone**

It is important to note that the control algorithm only uses the power input from the wind turbine when the PV insolation data is zero (night time); otherwise the PV is the primary power source. Keeping this in mind, this case performs better than the PV alone; however it cannot compare to the performance of the wind turbine alone due to the power output as previously discussed. The SOC ends up at about 60%, which is far better than the PV case which ended up at 0%. Of course, for the intelligent load control case the system SOC is maintained and the battery life is protected.

#### **4.6.6 PV, Wind Turbine, and Microgrid.**

As in the PV and wind turbine standalone case the primary power source is the PV. The Wind only becomes a power source when no PV power is present. In the NILC case above 80% SOC the microgrid and the primary loads draw power and the SOC quickly falls due to the insufficient PV power. Below 80% SOC only the primary loads drain power. In the case of the ILC the SOC is maintained by shutting off the microgrid and the primary loads when the SOC drops below 80%. The primary disadvantage of this control method is that the loads requesting power may not always receive it.

#### **4.6.7 Data Summary**

When looking at the summary data it becomes quite clear that unless you have a great deal of input power under favorable conditions, it is not recommended to run the system without intelligent load control over a long period of time. The only cases that performed well without intelligent load control were the cases where the wind turbine was the primary power source. However, given an undesirable set of wind data (i.e. no wind or very little) the results could easily be different. Therefore intelligent load control must be used in all cases.

If the system is to run in a long term scenario, it is shown that intelligent load control or some variation of it must be used in order to protect the battery from excessive discharge due to unforeseen input power or load variations. Therefore the best measure of the system performance in this case would be the cumulative amount of loads the system could service under intelligent load control. The results show that the wind turbine running as the primary power source is the best case (as well as the load profile) since it is able to service all the loads required of it while still maintaining the battery charge above 80%.

One important item to note is that due to intelligent load control and the extra microgrid loading the SOC of the battery is forced below the 80% set point much quicker than the cases where there is no microgrid. The intelligent load control will keep the loads off until the SOC rises above the 80% set point and this only usually occurs during charging. For the PV case charging will only occur the next morning when the sun rises. In the case of the PV and wind together, charging will occur the next morning or during nighttime if the wind is blowing. The disadvantage is that even though abundant wind

power may be during the day, the control system will default to the PV as the primary power source as long as a small amount of power is being generated from the PV

Due to the intelligent load control, the system without the microgrid actually performs better (in regard to load servicing) than the microgrid case. Even though there is more load for the system to service in the microgrid case, the intelligent load control is cutting off those loads faster and not servicing the loads until the next day. The non-microgrid case is not forced below the 80% SOC set point as fast and therefore it is able to service the loads for a longer period of time.

## **Chapter 5: Conclusion**

### **5.1 Achievements**

The bulk of the achievements made in this project are a direct result of many hours of research and careful structuring of the many components involved in the system. Future users of this model should have no trouble in simulating many different configurations and scenarios. Every effort was made to make the simulation easy to understand, alter, and adjust. All of the parameter adjustment and data recording is placed in one common area so that a person with no knowledge of the simulation could quickly run a given set of data. The simulation speed improvement is a true benchmark in terms of usability. Having a simulation that will tell you what may happen over the course of a week or more in fewer than 20 minutes is extremely useful.

Furthermore, multiple loads were added and some of the pre-existing loads were altered to increase model realism. A battery model was created which included the temperature effects and the SOC. Also, an initial intelligent load switching algorithm was written, which if implemented should enable the SuPER project to begin to start long term testing and data recording, while not endangering the battery. Another controller was written to observe the effects of using the T.E. cooler to maintain a temperature range in order to extend the life of the battery and decrease power consumption.

The initial design and creation of a DC microgrid model was to investigate long range power transmission of the SuPER system. Also, in order to investigate possibly adding a wind turbine to the system, a wind turbine model was created that was based directly on a commercially available option. A power source selection algorithm was also created to intelligently switch between the two power sources.

Several seven day simulations and the corresponding results were evaluated. The simulation was shown to be fast and error free. The model results conformed to what was expected; furthermore the simulation proved that it can handle multiple input data sets and configurations.

## **5.2 Future Recommendations**

### **5.2.1 T.E. Cooling for the Battery to Extend Battery Life.**

A good thing to investigate would be placing the battery inside a temperature controlled chamber, more specifically a temperature controlled chamber that is cooled via the thermoelectric effect. If the battery temperature is maintained at approximately 25°C, most of the experimental studies [3] indicate that the battery life can be doubled. Furthermore, I learned from my summer internship with ConocoPhillips that their entire battery bank (used for high voltage relay switching) is air conditioned to 75°F (24°C). Obviously, the company knows that keeping the battery's temperature controlled helps save them money on replacing batteries in the long run. Since the goal of the SuPER system is to have a mean time between failure rate of 25 years and currently the battery's anticipated life is only 5 years; it makes sense that the battery should be temperature controlled.

Another use for a temperature controlled chamber is to gather data to create a better model of the battery. The current battery temperature model is based on data that is not specific to the battery. If the system utilized a temperature controlled chamber, data could be taken under controlled conditions and therefore a more accurate battery model could be created.



### **5.2.2 Implement Parallel Processing for Speed Improvements in the Simulation.**

While the speed of the current simulation has been significantly improved over the previous version, there is still room for improvement. In the future someone may want to run a year's worth of data through this simulation to see how the system would perform. Using the current benchmark of 13 minutes for 7 days, a year would take  $52 \times 13 = 676$  minutes. This would be 11.3 hours of simulation time. It would be very helpful if the processing time could be sped up.

On the MathWorks website an add-on toolbox called Parallel Computing Toolbox 4.2 [49] was discovered. According to the website, the toolbox would allow the simulation to run on multiple processors. Also, the toolbox would allow use of the MATLAB's distributed Computing Server [37]. This server is similar to what S.E.T.I. does in order to process all of their deep space data [37]. A large simulation could be split up and the processing time can be distributed amongst thousands of computers. Use of this toolbox would allow SuPER team to run longer simulations in a short amount of time.

### **5.2.3 Frame Based Processing to Increase Simulation Speed**

There are a great number of free webinars on a large variety of topics at the MATLAB website. A particular webinar [38] discussed several different ways to speed up a Simulink model. Frame based processing was mentioned as a very appealing way to perform the simulation because it could decrease the simulation time by up to a factor of 30 [38]! Currently the simulation uses sample based processing, which essentially grabs and processes the data sample by sample.

Frame based processing essentially grabs a set of multiple samples and processes those all at once. Frame based processing was not implemented in the model because in order to do so each block (153 blocks) would have to be configured for frames. At this point in the simulation progress it would have taken up too much time.

#### **5.2.4 Battery Discharge Characteristic Tracking**

In order to obtain a more accurate mathematical model of the battery, it is recommended that the charge and discharge characteristics be recorded on a long term basis. Currently the maximum length of time the system has been run has been less than two days. More data is necessary so that the models of the system behavior can be refined both in the simulation and on the control system itself. Having a more accurate model of the system will allow maximum use of all the system resources and increase system efficiency.

#### **5.2.5 Increase DC-DC Converter Accuracy**

The first attempt at the DC-DC converter was not feasible because of the large number of differential elements and the small step size required by the PWM signal. If at some later date the simulation can be sped up even more, then re-implementing this model may be useful if someone wants to analyze the transient effects of the DC-DC converter.

#### **5.2.6 Implement in Software the Intelligent Load Control**

If the SuPER system is going to be run on a long term basis, it is necessary that intelligent load control be used. Allowing the system to operate simply based upon a timing schedule or human need will leave the system susceptible to excessive battery

discharge. If the state of charge is monitored and the system turns the loads on or off based on a safety set point, the battery can be protected from being excessively discharged. This algorithm can be easily setup since all the loads are already controlled via MOSFET switches. The primary thing to be aware of is that enough data must be gathered so that the SOC estimation algorithm is precisely matched to the battery; otherwise the system may prematurely turn off the loads or not turn off the loads soon enough.

### **5.2.7 Create a Better Power Source Selection Algorithm**

If a wind turbine is ever added to the Cal Poly SuPER system a power source selection algorithm will have to be written since only one power source can regulate the system voltage at any given time. While the power source selection algorithm presented in the simulation accomplished the goal, a much better algorithm could be written. Perhaps an algorithm that uses expected PV and wind speed data for a given day. The algorithm could switch to the PV or the wind based upon the expected input power from either source. However, the algorithm must be written to avoid fast switching between power sources because the switching may cause system instability.

## Bibliography

- [1] Sheffield, Tyler. "SuPER System Prototype Design and implementation." Master's thesis, California Polytechnic State University, April 2007. Web. September 11, 2009.  
<[http://courseware.ee.calpoly.edu/~jharris/research/super\\_project/ts\\_thesis.pdf](http://courseware.ee.calpoly.edu/~jharris/research/super_project/ts_thesis.pdf)>.
- [2] Oxtoby D.W., Freeman W.A., and Block, T.F. Chemistry Science of Change Third Edition. Saunders College Publishing, 1998.
- [3] Bode, Hans. Lead-Acid Batteries. John Wiley & Sons, Inc, 1997.
- [4] B.Wu, R.Dougal, and R.E. White. "Resistive companion battery modeling for electric circuit simulations." Journal of Power Sources. 93(2001): 186-200.
- [5] Rao R., Vrudhula,S., and Rakhmatov, D.N. "Battery Modeling for Energy-Aware System Design." IEEE Computer Society, December 2, 2003.
- [6] Kawasaki, K., Matsumura, S. Iwabu, K., Fujimura, N., and Iima, T. "Autonomous Dispersed Control System for Independent Microgrid." Electrical Engineering in Japan. Vol.166(2006): 1121-1127.
- [7] National Renewable Energy Resources Laboratory. "Solar Radiation Research Laboratory." Web. November 4, 2009.  
<[http://www.nrel.gov/midc/srrl\\_aocs/](http://www.nrel.gov/midc/srrl_aocs/)>.
- [8] NASA. "International Space Station Familiarization." Mission Operations Directorate Space Flight Training Division. July 31, 1998.
- [9] Yoko Amano and Hiroyuki Kajiwar. "Maximum Output Power Control System of Variable-Speed Small Wind Generators." Electrical Engineering in Japan, Vol. 165 No.1, 2008.
- [10] Obara, Shin'ya. "Power characteristics of a fuel cell microgrid with wind power generation." International Journal of Energy Research. 31:1064-1075, January 2007.
- [11] Coleman Inc. "Power Chill Iceless 12V Cooler instructions for use." Web. September 6, 2010.  
<<http://www.coleman.com/coleman/images/pdf/5640.pdf>>

- [12] ERT. “EL922512XXHYZ (T.E. Cooler Outer Fan) Datasheet.” . Web. October 20, 2010. <[http://www.easychinasupply.com/producttrade/last\\_pt\\_pro\\_open/845813.html](http://www.easychinasupply.com/producttrade/last_pt_pro_open/845813.html)>
- [13] Baoying Eletric Appliances Factory. “Thermal Protector JW6-2.” Web. December 13, 2009. <[http://jsaydq.en.alibaba.com/product/245290967-208127000/thermal\\_protector\\_JW6\\_2\\_VDE\\_UL\\_with\\_a\\_casing\\_pipe\\_.html](http://jsaydq.en.alibaba.com/product/245290967-208127000/thermal_protector_JW6_2_VDE_UL_with_a_casing_pipe_.html)>.
- [14] TELLUREX. “Introduction to thermoelectrics and design manual.” Web. May 6, 2010. <<http://www.tellurex.com/technology/design-manual.php>>.
- [15] TELLUREX. “FAQ about cooling and heating technology.” Web. March 26, 2010. <<http://www.tellurex.com/pdf/peltier-faq.pdf>>.
- [16] MK Battery. “8G31-DEKA Battery Datasheet.” Web. May 16, 2009. <<http://www.mkbattery.com/images/8G31-DEKA.pdf>>.
- [17] Szymborski, Joseph. Battery Training Seminar. October 28, 2010.
- [18] Van Buskirk, Robert. Personal interview. February 14, 2009.
- [19] Harris, James G. Personal Photo. March 11, 2009.
- [20] eForCity. “Cell Car Charging Information.” Web. January 28, 2009. <<http://www.eforcity.com/blackberry-pearl-8100-car-charger-oem-asy09824001-black-cbla8100cc01.html?efprcshzldtf090605=cbla8100cc01>>.
- [21] AxionTech.Com. “Sangean PR-D7 Rechargeable AM/FM Radio.” Web. January 28, 2009. <<http://www.axiontech.com/prdt.php?item=88188>>.
- [22] “12V LED Bulbs EO EL16 A Specs.” *The LED Light.com*. Web. May 7, 2010. <<http://www.theledlight.com/led-fixtures.html> >.
- [23] Harris, James G. White Papers for Sustainable Power for Electrical Resources- SuPER. July 15, 2005. Web. July 15, 2008. <[https://docs.google.com/viewer?url=http://courseware.ee.calpoly.edu/~jharris/research/super\\_project/white\\_paper\\_susper.pdf](https://docs.google.com/viewer?url=http://courseware.ee.calpoly.edu/~jharris/research/super_project/white_paper_susper.pdf)>.
- [24] United States. NATIONAL ELECTRICAL CODE 2005 Edition. Table 310.16 page 70-148.
- [25] Andreyckak, Bill H. Zero Voltage Switching Resonant Power Conversion. Texas Instruments Incorporated, 2001.

- [26] BP Solar. "BP SX150- 150Watt Multicrystalline Photovoltaic Module Datasheet." Web. August 5, 2010.  
<<http://www.southwestpv.com/DOWNLOAD.HTM>>.
- [27] Southwest Windpower. "Air X Technical Specifications." Web. March 24, 2009. <[http://www.windenergy.com/documents/spec\\_sheets/3-CMLT-1339-01\\_Air\\_X\\_Spec.pdf](http://www.windenergy.com/documents/spec_sheets/3-CMLT-1339-01_Air_X_Spec.pdf)>.
- [28] Southwest Windpower. "Air X User Manual." Web. January 24, 2009.  
<<http://www.allsmallwindturbines.com/files/airx.pdf>>.
- [29] H.Yu, C.Y. Chung, K.P. Wong, and J.H. Zhang. "A Chance Constrained Transmission Network Planning Method With Consideration of Load and Wind Farm Uncertainties." IEEE Transactions on Power Systems, VOL 24 NO.3 (August 2009).
- [30] Ridgely, John. "Cal Poly Wind data." Email to the author. October 29, 2009.
- [31] Outback Power Systems, Inc. "MX60 Data Sheet." Web. August 29, 2009.  
<[http://www.dcpower-systems.com/uploads/products/19108\\_1.pdf](http://www.dcpower-systems.com/uploads/products/19108_1.pdf)>.
- [32] Witts, Joseph. "Cal Poly SuPER System System Photovoltaic Array Universal DC-DC Step Down Converter." Master's thesis, California Polytechnic State University, June 2008. Web. July 10, 2009.  
<[https://docs.google.com/viewer?url=http://courseware.ee.calpoly.edu/~jharris/research/super\\_project/jw\\_thesis.pdf](https://docs.google.com/viewer?url=http://courseware.ee.calpoly.edu/~jharris/research/super_project/jw_thesis.pdf)>.
- [33] Staniszewski, Matt. "Implementing a SPI Driver in uCLINUX for Cal Poly SuPER Project." Senior project, California Polytechnic State University, June 2009. Web. August 2, 2009.  
<[http://courseware.ee.calpoly.edu/~jharris/research/super\\_project/ms\\_Senior\\_Project\\_Report.pdf](http://courseware.ee.calpoly.edu/~jharris/research/super_project/ms_Senior_Project_Report.pdf)>.
- [34] HUGHES Network Systems. "Hughes DW7000 Satellite modem." Web. September 1, 2010. <<http://www.copperhead.cc/DW7000I.pdf>>.
- [35] CISCO systems. "Range Plus-n Wireless Router Datasheet." Web. December 12, 2009.  
<[http://downloads.linksysbycisco.com/downloads/WRT110\\_DS\\_V10\\_C.0.pdf](http://downloads.linksysbycisco.com/downloads/WRT110_DS_V10_C.0.pdf)>.
- [36] Witts, Joseph. SuPER Cart DC Motor Model and Ultra-Capacitor Addition. 4+1 Interim Report, California Polytechnic State University.

March 1, 2008. Web. September 10, 2009.  
<[http://courseware.ee.calpoly.edu/~jharris/research/super\\_project/jw\\_spring08\\_report.pdf](http://courseware.ee.calpoly.edu/~jharris/research/super_project/jw_spring08_report.pdf)>.

- [37] MATHWORKS. “Parallel Computing Toolbox 4.2 description.” . Web. May 7, 2010. <<http://www.mathworks.com/products/parallel-computing/>>.
- [38] Rodriguez, Alex and David Jackson. “Tips for Speeding up Simulink Models for Signal Processing and Communications (MATHWORKS webinar).” Web. May 7, 2010.  
<[http://www.mathworks.com/webex/recordings/SL\\_Tips/index.html](http://www.mathworks.com/webex/recordings/SL_Tips/index.html)>.

## Appendix A: C-MEX S-Function Code

### A.1 Battery Model S-Function Improved Code

```
/*
Written by Matthew O. McFarland
5/10/2009
*/

//Discharge Variables-----
double Kt_dis=1.1;
double Kc_dis=500;
double T_C;
double T_adj;
double T_const_high=.98;//1.5
double T_const_low=.98;
double temp_d_h=50;
double temp_d_l=50;

//Global Variables-----
int charge_flag;
double I;
double I_int;
double vout;
double Voc=13.8; //Open circuit voltage
double C=6480; //The battery capacity in amp*min
double Ve=10.5;
double K_SOC=150;

//Charge Variables-----
double chg_log=3.81;
double chg_exp=.037;

//MAIN CODE-----

//DETERMINE CHARGE MODE-----
if(Current[0]>=0){
    //We are in Discharge mode(Current is leaving the battery)
    charge_flag=0; //Set Charge Flag to 0 (not charging)
}
else{//We are in Charge mode (Current is entering the battery)
    charge_flag=1; //Set Charge Flag to 1 (charging)
}

//TAKE IN DATA-----
```



```

T_C=Temperature_diff[0]; //Measured Temperature in C
T_C=T_C+273.15;//CONVERT TO KELVIN
I=Current[0]; //Current in amps
I_int=Current_Integral[0];//Integral of current

//TEMPERATURE ADJUSTMENT EQUATION-----

if(Temperature_diff[0]==0){ //Temperature is equal to 25C
    T_adj=1;
}
else if (Temperature_diff[0]>0){ //Temperature is greater than 25C
    T_adj=(fabs(T_C)-temp_d_h)/(fabs(T_C)*T_const_high);
}
else if (Temperature_diff[0]<0){ //Temperature is less than 25C
    T_adj=(fabs(T_C)+temp_d_l)/(fabs(T_C)*T_const_low);
}

//DISCHARGE MODE-----
if(charge_flag==0){

    vout=Voc-((Kt_dis*C*T_adj)/(C-I_int))+Ve*exp(-Kc_dis*I_int);

}
//CHARGE MODE-----
else{

    if((1/((Kt_dis*C)/(C-I_int)))<.50){
        //This is where the charge curve is logarithmic in nature

        vout=2*log(chg_log*100*(1/((Kt_dis*C)/(C-I_int))));
    }
    else{
        //This is where the charge curve is exponential in nature
        vout=2*exp(chg_exp*100*(1/((Kt_dis*C)/(C-I_int))));
    }

}

Voltage_Out[0]=vout;
SOC[0]=K_SOC*(1/((Kt_dis*C*T_adj)/(C-I_int))); //CALCULATE SOC
MODE[0]=charge_flag;

```

## A.2 DC Microgrid S-Function Code

```

//The output voltage for the HVDC link will be held at 160VDC

```

```
//Link Shutdown conditions:
// 1)SOC of the battery drops below 60%
// 2)The Power Consumed by the link exceeds the Power consumptions setpoint
// 3)A fault condition exists indicated by an above average current.
// 4)Link Shutdown terminal is asserted LOW (0)
```

```
//Link Reactivation Conditions:
// 1) SOC of the battery is above 60%
// 2) The Power consumed< Power consumption setpoint
// 3) No fault condition
// 4) Link shutdown terminal is asserted HIGH(1)
```

```
double power_consumption_setpoint=70;//Set Power Drain Limit 70W To drain
double i_out_fault_setpoint=.75; // Fault current setpoint 750mA
double soc_de_activate=60; //SOC limit to de-activate link
double vout_base=160;//Base Trasmission voltage 160VDC
```

```
double power_consumed_int;
```

```
power_consumed_int=vout_base*i_out_dc_link[0]; //This is the instantaneous
power consumed from the link
```

```
//Perform the tests:
```

```
if(fault_flag_status[0]==0) //No Fault is set
{
    if((SOC_batt[0]>soc_de_activate) && (Link_control_on_off[0]==1) && (
power_consumed_int < power_consumption_setpoint ) &&
(i_out_dc_link[0]<i_out_fault_setpoint) ) //Passes All Tests
    {
        V_HVDC_OUTPUT[0]=vout_base; //Turn on the link
        I_current_source_batt_side[0]=power_consumed_int/V_batt_bus[0]; //This
sets the input current to reflect the output current being drained
        Fault_Flag[0]=0;
    }
    else //Fails SOC test, Link Test, Power Consupntion test, and Fault current test
    {
        V_HVDC_OUTPUT[0]=0; //Shuts Down LINK
        Fault_Flag[0]=1; //Sets the Fault Flag
    }
}
else if(cnt[0]>10) // Fault Flag is set and count has exceeded 10 counts
{
    if((SOC_batt[0]>soc_de_activate) && (Link_control_on_off[0]==1) && (
power_consumed_int < power_consumption_setpoint ) &&
(i_out_dc_link[0]<i_out_fault_setpoint) ) //Passes All Tests
```

```

{
    V_HVDC_OUTPUT[0]=vout_base; //Turn on the link
    I_current_source_batt_side[0]=power_consumed_int/V_batt_bus[0]; //This
sets the current drained from battery to reflect the current being drained power
being drained through the HVDC link
    Fault_Flag[0]=0; //Sets the fault flag low
}
else //Fails SOC test, Link Test, Power Consumption test, and Fault current test
{
    V_HVDC_OUTPUT[0]=0; //Shuts Down LINK
    Fault_Flag[0]=1; //Sets the Fault Flag
}
}
else //count is less than 10 and therefore keep dc microgrid off
{
    V_HVDC_OUTPUT[0]=0;
    I_current_source_batt_side[0]=0;
}
//Given that  $P=VI \Rightarrow$  using a current source as a variable resistor  $I=P/V_{\text{measured}}$ 
I_current_source_batt_side[0]=power_consumed_int/V_batt_bus[0];

```

### A.3 Hut Scenario Load Control Block S-Function Code

```

// This program was written by Matthew O. McFarland
// 4/19/2009

// LOGIC 0= OFF for the associated load
// LOGIC 1= ON for the associated load
// Time input units: Hours

// Total Time for a day: 24 hrs
// Time of Sunrise: 0800
// Time of Sunset: 1800

//Array Initialization//
int RADIO[1][24]= {
    {0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0},//SCENARIO
};

int CELL[1][24]= {
    {0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0},//SCENARIO
};

int LEDS[1][24]= {
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0},//SCENARIO
};

```

```
//OUTPUT STATMENTS
```

```
GATE_RADIO[0]= RADIO[1][TIME[0]];
```

```
GATE_CELL[0]= CELL[1][TIME[0]];
```

```
GATE_LEDS[0]= LEDS[1][TIME[0]];
```

#### A.4 Wind Turbine S-Function Code

```
double inst_power_available; //Instantaneous Power Available
double vreg; //Voltage Regulation Setpoint
double vcut_in; //If the batt/bus voltage drops to this point then
                //If power is available Vreg will be placed onto the bus
double i_limit;

vreg=13.63;
vcut_in=11.5;

//The wind Power curves
if (Vwmph[0]<30)
{
    inst_power_available=.0312*pow(Vwmph[0],3)-
    .04855*pow(Vwmph[0],2)+4.8266*Vwmph[0];//-0.6729;
}
else if (Vwmph[0]>30)
{
    inst_power_available=151.64*log(Vwmph[0])-486.39;
}
//Now that we know the instantaneous power available,
//The Intelligent microcontroller increase the voltage output
//To charge the battery until the battery voltage matches Vreg
//Once the Vreg is reached the microcontroller will turn off the output
//Voltage and will remain off until the vcut_in is reached, then the Battery
//will be charged again.
//Ref Air Breeze Technical Specification

//We know that  $P=I*V \Rightarrow$  for any given  $V$   $I=P/V$ 
//This will be our limit, if I exceed this value then the generator
//will turn off

i_limit=inst_power_available/vreg;
```

```

    if((I_delivered[0]>i_limit) && (Vwmp[0]!=0)) //No Wind or Too much current
delivered
    {
        Vout[0]=0;
    }
    else //We are within limits
    {
        if(V_bus[0]<vcut_in)
        {
            Vout[0]=vreg; //Supply the correct bus voltage
        }
        else if(V_bus[0]>vreg)
        {
            Vout[0]=0; //Turn off the voltage source we have reached setpoint
        }
        else
        {
            Vout[0]=vreg; //Supply the correct bus voltage
        }
    }
}

```

## A.5 PV DC-DC Converter Control S-Function Block Code

```

//VARIABLE DECLARATIONS-----
double p_available;
double R_set=9;
double p_deliever;
double IMAXOUT;

//CALCULATE POWER AVAILABLE AT THE INPUT-----
p_available=(V_IN[0]*V_IN[0])/R_set;

//Determine the actual amount of power you can deliver-----
p_deliever=(8E-10)*pow(p_available,5)-(4E-10)*pow(p_available,4)+
(8E-5)*pow(p_available,3)-.0083*pow(p_available,2)+.3503*p_available+90.827;

//Now Determine the maximum current you can put out of the DC-DC Converter
//FOR THE GIVEN Duty cycle, Voltage, and Power
//Since P=I*V ==> Imax=p_deliever/V_out
//Vout=DVin

IMAXOUT=p_deliever/(DUTY_CYCLE[0]*V_IN[0]);

```

```

if(I_OUT[0]>IMAXOUT){//OUTPUTTING TOO MUCH CURRENT FOR GIVEN
SETTINGS
    V_OUT[0]=0;
}
else{ //WE ARE WITHIN LIMITS
    V_OUT[0]=DUTY_CYCLE[0]*V_IN[0];
}

```

## A.6 PV Charge Control Block S-Function Code

```

/*
Written by Matthew O. McFarland
5/10/2009
Code reflects the code as configured with the laptop
*/

////////////////////VARIABLE DECLARATIONS////////////////////
double Ppv_new;

double delta_Pv;

double DcStep=.0025; //Sets the duty cycle step size
double DcInitStep=.05; //Set the initial duty cycle step size

double maxVset=14.5; //Sets the maximum voltage to charge to

double floatcharge_set=14.5; //Sets when to start a float charge
double floatcharge_volt=13.9;

double startcharge_set=13.8; //Sets when to start a normal charge
double startcharge_volt=13.5;

double duty_calc;

////////////////////MAIN BODY OF THE PROGRAM////////////////////

Ppv_new = Vpv[0]*Ipv[0]; //Calculate Array Power

//Want to charge the battery using predefined setpoints.

if((Ppv_new<0)|| (Vbatt[0]>maxVset)){ //No Power Available or Max setpoint
    duty_cycle[0]=0; //==> Turn off duty cycle
}

```

```

else{ //POWER IS AVAILIBLE

    if(Vbatt[0]<startcharge_set){//If the voltage drops to this level
        //float charging starts
        //Calculate the necessary duty cycle to achieve our voltage
        //D=Vout/Vpv

        if(startcharge_volt*Iout[0]>Ppv_new){ //Power trying to output is
            //greater than input
            duty_cycle[0]=DC_PREV[0]-DcStep;
        }
        else{
            duty_calc=startcharge_volt/Vpv[0];
            duty_cycle[0]=duty_calc;
        }

    }
    else if (Vbatt[0]< floatcharge_set){//If the voltage drop to this level
        // charge starts
        if(floatcharge_volt*Iout[0]>Ppv_new){ //Power trying to output is
            //greater than input
            duty_cycle[0]=DC_PREV[0]-DcStep; //Lower Duty Cycle
        }
        else{
            duty_calc=floatcharge_volt/Vpv[0];
            duty_cycle[0]=duty_calc;
        }
    }
}

}

```

### A.7 Motor Model Control Block S-Function Block Code

```

double iout;

if(Motor_Gate_Sig[0]==0){ //THIS IS THE OFF CASE

    STOP_CNT[0]=0; //HALTS COUNTER, HOLDS COUNTER AT ZERO
    RESET_SIG[0]=1; //RESETS THE COUNTER
    iout=.00000001; //Essentially an open
    lcontrol[0]=iout;
}
else{//THE MOTOR IS TURNED ON

    STOP_CNT[0]=1; //ENABLES COUNTING

```

```

RESET_SIG[0]=0;//TURNS OFF THE RESET SIGNAL

if(Time_Count[0]<7){//THIS IS THE STARTUP TIME

//Calculates Iout during startup 0-1000ms
//Iout=40; //SIMULATION SAMPLE TIME IS CURRENTLY SET TO LOW
//TO REFLECT THE TRANSIENT EFFECTS with the equation listed below
Iout=(-.0001)*pow(Time_Count[0],5)+(.0065)*pow(Time_Count[0],4)-
(.1412)*pow(Time_Count[0],3)+(1.5467)*pow(Time_Count[0],2)-
(8.6405)*Time_Count[0]+ 40.046;
}

else{//THIS IS THE STEADY STATE
Iout= 20;

STOP_CNT[0]=0; //TURNS OFF COUNTER
//RESET_SIG[0]=1; //RESETS THE COUNTER
}
Icontrol[0]=Iout;
}

```

#### **A.8 T.E. Cooler Thermal Model S-Function Code**

```

//THIS MODEL WAS WRITTEN BY MATTHEW O. MCFARLAND
//BASED ON EXPERIMENTAL DATA AS TAKEN BY TYLER SHEFFIELD
//THE THERMAL MASS IS 6 QUARTS OF WATER AND THE EXTERNAL
TEMPERATURE
//WAS AROUND 70 F
//THE HEATING AND COOLING RATES WERE FOUND AT THIS SPECIFIC
TEMPERATURE
//KEEP IN MIND THAT THE HEATING AND COOLING RATES WILL BE
DIFFERENT
//FOR DIFFERENT AMBIENT TEMPERATURES

//Only one assumption was made that did not correlate exactly according the
//data:
//TE ON: Tamb<Tint: Tint=Tint-K2-K3*deltaC {K2-from cooler K3-from cond heat
loss}
//    Tamb>Tint: Tint=Tint-K2

//TE OFF: Tamb<Tint: Tint=Tint-K3*deltaC {K1-from heat up K3-from cond heat}
//    Tamb>Tint: Tint=Tint+K1

static int execute;
static double Tint_C;

```



```

double TambiC;

double K1=0.0044;//Based upon experimental data (HEAT UP)
double K2=0.075;//Based upon experimental data (COOLDOWN)
double K3=.00044; //Made identicle to heat up rate (COOLDOWN insulation loss)

TambiC = TambC[0]; //PLACE INPUT INTO VARIABLE

TINTUPPER[0]=TambC[0]+100; //Essentially Upper is unbounded
TINTLOWER[0]=TambC[0]-25; //THIS IS THE MAXIMUM THE TE COOLER
CAN COOL TO

if (execute<=0){ //THIS SHOULD ONLY EXECUTE ONCE
    //TAKES IN INITIAL Tamb and Tcooler
    Tint_C = TintC[0];
    execute=2;
}

if(Gate_Sig[0]==1){//DEVICE IS ON

    if( TambiC > Tint_C ){ //Tambient greater that Tint
        Tint_C = Tint_C - K2 ; //HEAT LOSS EQUATION
    }
    else if (TambiC < Tint_C){ //Tambient less than Tint
        Tint_C = Tint_C -K2-K3 ; //HEAT LOSS EQUATION
    }
}

else{ //DEVICE IS OFF

    if(TambiC > Tint_C){ //Tambient greater that Tint
        Tint_C = Tint_C + K1; //HEAT GAIN EQUATION
    }
    else if (TambiC < Tint_C){ //Tambient less than Tint
        Tint_C = Tint_C - K3; //HEAT LOSS EQUATION
    }
}

TintC_out[0] = Tint_C;

```

### A.9 T.E. Cooler Electric Model S-Function Code

//BASED UPON EXPERIMENTAL DATA AS TAKEN BY TYLER SHEFFIELD.  
 //An exact mathematical relationship between the system temperatures

```
//And the power consumed by the T.E. cooler could not be obtained
//Therefore the average power output of a cooler in an on state
//Was made to be the power consumed.
//In the future this model can be altered to create a more realistic model.
```

```
double iout;
```

```
if(Gate_Sig[0] > 0){ //GATE SIGNAL ASSERTED==>TURNED ON
```

```
    iout=5;//Constant 60W consumed
}
```

```
else{
    iout = .0000001 ;//Essentially makes it an open
}
```

```
i_control[0] = iout; //Assigns the output to the port
```

#### **A.10 Scenario Load Control S-Function Block Code**

```
// This program was written by Matthew O. McFarland
// 4/19/2009
```

```
// LOGIC 0= OFF for the associated load
// LOGIC 1= ON for the associated load
// Time input units: Hours
```

```
// Total Time for a day: 24 hrs
// Time of Sunrise: 0800
// Time of Sunset: 1800
```

```
//Array Initilization//
```

```
int tv_array[6][24]= {
    {0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0},//SCENARIO 1
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 2
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 3
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 4
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 5
    {0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0},//SCENARIO 6
};
```

```
int cooler_array[6][24]={
    {0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0},//SCENARIO 1
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 2
    {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},//SCENARIO 3
    {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},//SCENARIO 4
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 5
};
```

```

    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 6
};

```

```

int led_array[6][24]={
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0},//SCENARIO 1
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1},//SCENARIO 2
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1},//SCENARIO 3
    {0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1},//SCENARIO 4
    {0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1},//SCENARIO 5
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1},//SCENARIO 6
};

```

```

int motor_array[6][24]={
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0},//SCENARIO 1
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0},//SCENARIO 2
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 3
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 4
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 5
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 6
};

```

```

int internet_array[6][24]={
    {0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0},//SCENARIO 1
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 2
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 3
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 4
    {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},//SCENARIO 5
    {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},//SCENARIO 6
};

```

//OUTPUT CODE

```

if(SOC[0]>80){//THE SOC IS ABOVE 80 USE NORMAL LOAD SCENARIOS
TV_Control[0]=tv_array[scenario_num[0]][time[0]];
Cooler_Control[0]=cooler_array[scenario_num[0]][time[0]];
LED_Control[0]=led_array[scenario_num[0]][time[0]];
Motor_Control[0]=motor_array[scenario_num[0]][time[0]];
Internet_Control[0]=internet_array[scenario_num[0]][time[0]];
}
else{ //THE SOC IS BELOW 80 TURN OFF ALL LOADS
TV_Control[0]=0;
Cooler_Control[0]= 0;
LED_Control[0]=0;
Motor_Control[0]= 0;
Internet_Control[0]=0;
}

```

## Appendix B: Wind Data Processing Macros

### B.1 Wind Data Raw Processing Macro (Written in VBA)

```
Sub Macro1()  
  
Dim I, j, k As Integer  
Dim variable As String  
Dim Length As Integer  
Dim start As Integer  
Dim num_of_char As Integer  
  
Dim time_stamp As String  
Dim first_char As String  
Dim test_char As String  
  
Dim length_of_avg_wind_speed As Integer  
Dim length_of_sensor_num As Integer  
Dim sensor_num As Integer  
Dim avg_wind As Integer  
Dim cnt As Integer  
  
Dim sensor_num_str As String 'Testing Variable  
  
Dim space(1, 10) As Integer  
  
k = 1  
cnt = 2  
  
Sheets("Sheet1").Select  
Range("A" & 1).Value = "TIME STAMP"  
Range("B" & 1).Value = "AVG WIND SPEED"  
Sheets("wind-today").Select  
For I = 1 To 2577 '  
    variable = Range("A" & I).Value  
    Range("A" & I).Select  
    'Now we need to parse up the text in the variable  
    'We need to search for the ^ character  
    'Once we find the variable we need to parse up the data and store it into variables  
    'Then store those values onto different sheet entries  
  
    first_char = Mid(variable, 1, 1) 'Grabs the first character in the string  
    If first_char = "^" Then ' The start character  
        Length = Len(variable)  
  
        For j = 1 To Length 'Search for all occurrences of the space character  
            'Typical format is ^02:30:41_18_3_16_0_0_0 There should be 7 spaces  
            'Start Char_Time_Sensor#_AvgWind_MAXWIND_MINWIND_DIRECT_DIRECT  
  
            test_char = Mid(variable, j, 1)  
            If test_char = " " Then 'If a space occurs then record where it occurs  
                space(1, k) = j  
                k = k + 1  
            End If  
        Next j  
        k = 1 'Resets after it exits this  
  
        'Now that we know where the spaces are we can parse up the data according  
        'To the order  
        '8 characters past the first space is the time stamp  
        'space character position 4 - space character position 3 is the length of the sensor #  
        'space char pos 5 - space char pos 4 is the length of the avg speed  
        'Space char pos 4 +avg speed length is the avg speed  
  
        length_of_sensor_num = space(1, 4) - space(1, 3)  
  
        length_of_avg_wind_speed = space(1, 5) - space(1, 4)  
  
        time_stamp = Mid(variable, space(1, 1), 9)
```

```

        sensor_num = Val(Mid(variable, space(1, 3), length_of_sensor_num))

        avg_wind = Val(Mid(variable, space(1, 4), length_of_avg_wind_speed))

        If sensor_num = 18 Then 'This is the data we want to use change to get different sensor#
            Sheets("Sheet1").Select 'Switch to the other sheet
            Range("A" & cnt).FormulaR1C1 = time_stamp 'Write the data
            Range("B" & cnt).Value = avg_wind
            Sheets("wind-today").Select
            cnt = cnt + 1
        End If
    End If
    For j = 1 To 10
        space(1, j) = 0 'This will clear the array after using it
    Next j
Next I

End Sub

```

## B.2 Wind Data All Seven Days Algorithm (Written in VBA)

```

Sub Macro1()
'
' Macro1 Macro
' Macro recorded 1/16/2010 by Matthew McFarland
'
Dim last_row As Integer
Dim I As Integer
Dim j As Integer
Dim cell_value As String

Dim start_row As Integer

Dim directory As String
Dim file_name1 As String
Dim file_name2 As String
Dim file_name3 As String
Dim file_name4 As String
Dim file_name5 As String
Dim file_name6 As String
Dim file_name7 As String

Dim column_to_look_at As String
Dim cell_contents As String

column_to_look_at = "B"
cell_contents = "asdfa" 'a non "" value
directory = "C:\Documents and Settings\Owner\Desktop\School\Thesis\Wind Turbine Modeling\
file_name1 = "4_7_2009.xls"
file_name2 = "4_8_2009.xls"
file_name3 = "4_9_2009.xls"
file_name4 = "4_10_2009.xls"
file_name5 = "4_11_2009.xls"
file_name6 = "4_12_2009.xls"
file_name7 = "4_13_2009.xls"

```

```

'FILE1-----
Workbooks.Open Filename:=directory & file_name1 ' Open The File

Windows(file_name1).Activate 'Select the sheet

Range("B2:B" & 289).Select 'Select all data up to the last row
'***THERE ARE ONLY 288 minute intervals per day*****
|
Selection.Copy 'Copy the data up to the last row

Windows("ALL_7_DAYS.xls").Activate 'Go to the All data sheet

start_row = 2

Range(column_to_look_at & start_row).Select 'Pastes to this

ActiveSheet.Paste

'Write the numbers into the time box

column_to_look_at = "A"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    Range(column_to_look_at & (j + start_row)).value = I
    j = j + 1
Next I

column_to_look_at = "B"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    cell_value = Range("B" & (j + start_row))
    If cell_value = "" Then
        Range("B" & (j + start_row)).value = 0
    End If
    j = j + 1
Next I

'Close the file
    Windows(file_name1).Activate
    ActiveWorkbook.Close
'FILE2-----
Workbooks.Open Filename:=directory & file_name2 ' Open The File

Windows(file_name2).Activate 'Select the sheet

Range("B2:B" & 289).Select 'Select all data up to the last row
'***THERE ARE ONLY 288 minute intervals per day*****

Selection.Copy 'Copy the data up to the last row

Windows("ALL_7_DAYS.xls").Activate 'Go to the All data sheet

start_row = 290

Range(column_to_look_at & start_row).Select 'Pastes to this

ActiveSheet.Paste
|
'Write the numbers into the time box

```

```

column_to_look_at = "A"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    Range(column_to_look_at & (j + start_row)).value = I + 288
    j = j + 1
Next I

column_to_look_at = "B"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    cell_value = Range("B" & (j + start_row))
    If cell_value = "" Then
        Range("B" & (j + start_row)).value = 0
    End If
    j = j + 1
Next I

'Close the file
Windows(file_name2).Activate
ActiveWorkbook.Close
'FILE3-----
Workbooks.Open Filename:=directory & file_name3 ' Open The File

Windows(file_name3).Activate 'Select the sheet

Range("B2:B" & 289).Select 'Select all data up to the last row
'***THERE ARE ONLY 288 minute intervals per day*****

Selection.Copy 'Copy the data up to the last row

Windows("ALL_7_DAYS.xls").Activate 'Go to the All data sheet

start_row = 578

Range(column_to_look_at & start_row).Select 'Pastes to this

ActiveSheet.Paste

'Write the numbers into the time box

column_to_look_at = "A"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    Range(column_to_look_at & (j + start_row)).value = I + 288 * 2
    j = j + 1
Next I

column_to_look_at = "B"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    cell_value = Range("B" & (j + start_row))
    If cell_value = "" Then
        Range("B" & (j + start_row)).value = 0
    End If
    j = j + 1
Next I

'Close the file
Windows(file_name3).Activate
ActiveWorkbook.Close
'FILE4-----
Workbooks.Open Filename:=directory & file_name4 ' Open The File

Windows(file_name4).Activate 'Select the sheet

```

```

Range("B2:B" & 289).Select 'Select all data up to the last row
'***THERE ARE ONLY 288 minute intervals per day*****

Selection.Copy 'Copy the data up to the last row

Windows("ALL_7_DAYS.xls").Activate 'Go to the All data sheet

start_row = 866

Range(column_to_look_at & start_row).Select 'Pastes to this

ActiveSheet.Paste

'Write the numbers into the time box

column_to_look_at = "A"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    Range(column_to_look_at & (j + start_row)).value = I + 288 * 3
    j = j + 1
Next I

column_to_look_at = "B"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    cell_value = Range("B" & (j + start_row))
    If cell_value = "" Then
        Range("B" & (j + start_row)).value = 0
    End If
    j = j + 1
Next I

'Close the file
    Windows(file_name4).Activate
    ActiveWorkbook.Close

'FILES-----
Workbooks.Open Filename:=directory & file_name5 ' Open The File

Windows(file_name5).Activate 'Select the sheet

Range("B2:B" & 289).Select 'Select all data up to the last row
'***THERE ARE ONLY 288 minute intervals per day*****

Selection.Copy 'Copy the data up to the last row

Windows("ALL_7_DAYS.xls").Activate 'Go to the All data sheet

start_row = 1154

Range(column_to_look_at & start_row).Select 'Pastes to this

ActiveSheet.Paste

'Write the numbers into the time box

column_to_look_at = "A"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    Range(column_to_look_at & (j + start_row)).value = I + 288 * 4
    j = j + 1
Next I

column_to_look_at = "B"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    cell_value = Range("B" & (j + start_row))
    If cell_value = "" Then
        Range("B" & (j + start_row)).value = 0
    End If
    j = j + 1

```



```

Next I

'Close the file
    Windows(file_name5).Activate
    ActiveWorkbook.Close

'FILE6-----
Workbooks.Open Filename:=directory & file_name6 ' Open The File

Windows(file_name6).Activate 'Select the sheet

Range("B2:B" & 289).Select 'Select all data up to the last row
'***THERE ARE ONLY 288 minute intervals per day*****

Selection.Copy 'Copy the data up to the last row

Windows("ALL_7_DAYS.xls").Activate 'Go to the All data sheet

start_row = 1442

Range(column_to_look_at & start_row).Select 'Pastes to this

ActiveSheet.Paste

'Write the numbers into the time box

column_to_look_at = "A"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    Range(column_to_look_at & (j + start_row)).value = I + 288 * 5
    j = j + 1
Next I

column_to_look_at = "B"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    cell_value = Range("B" & (j + start_row))
    If cell_value = "" Then
        Range("B" & (j + start_row)).value = 0
    End If
    j = j + 1
Next I

'Close the file
    Windows(file_name6).Activate
    ActiveWorkbook.Close

'FILE7-----
Workbooks.Open Filename:=directory & file_name7 ' Open The File
|
Windows(file_name7).Activate 'Select the sheet

Range("B2:B" & 289).Select 'Select all data up to the last row
'***THERE ARE ONLY 288 minute intervals per day*****

Selection.Copy 'Copy the data up to the last row

Windows("ALL_7_DAYS.xls").Activate 'Go to the All data sheet

start_row = 1730

Range(column_to_look_at & start_row).Select 'Pastes to this

ActiveSheet.Paste

'Write the numbers into the time box

column_to_look_at = "A"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    Range(column_to_look_at & (j + start_row)).value = I + 288 * 5
    j = j + 1
Next I

```

```

column_to_look_at = "B"
j = 0
For I = 1 To 288
    Range(column_to_look_at & (j + start_row)).Select
    cell_value = Range("B" & (j + start_row))
    If cell_value = "" Then
        Range("B" & (j + start_row)).value = 0
    End If
    j = j + 1
Next I

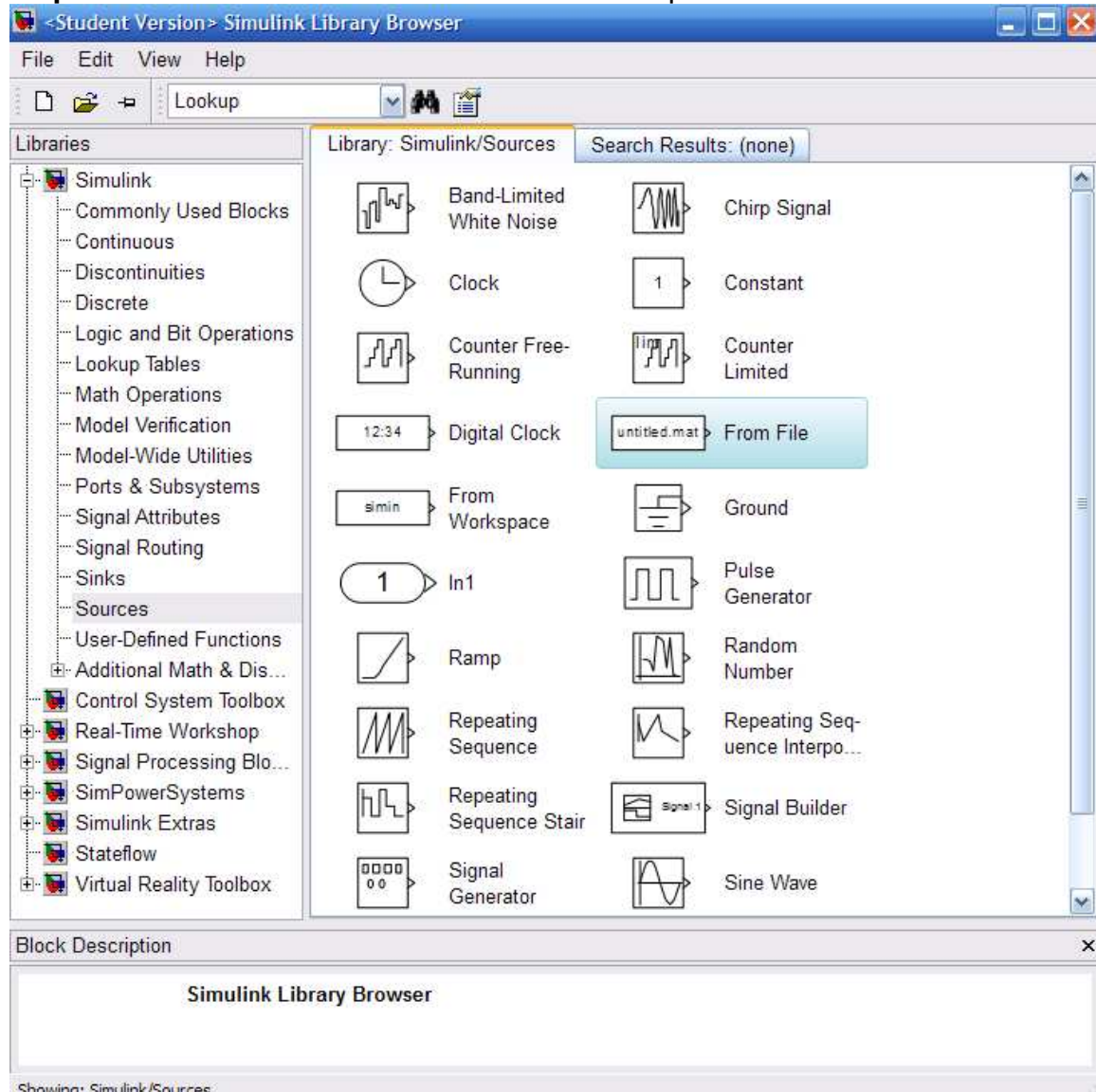
'Close the file
Windows(file_name7).Activate
ActiveWorkbook.Close
End Sub

```

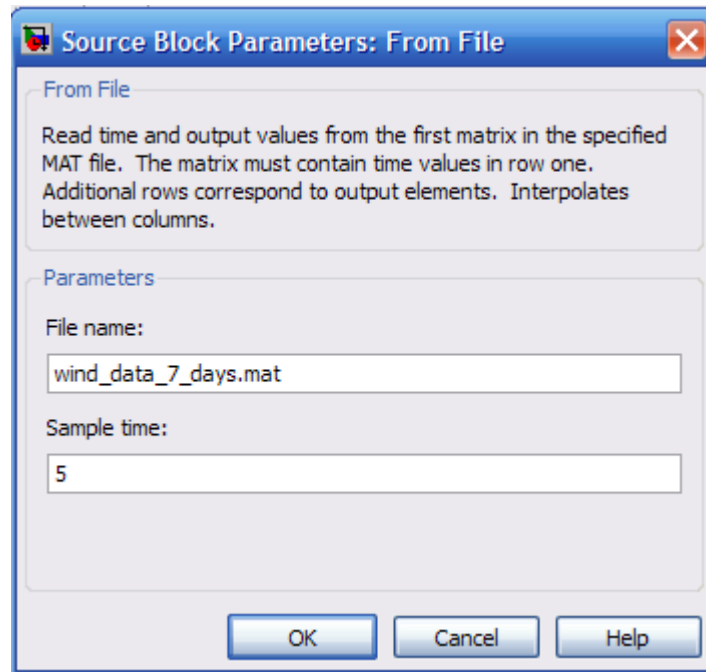
## Appendix C: MATLAB/SIMULINK Data Manipulation Commands and Functions

### C.1 Using From File Block

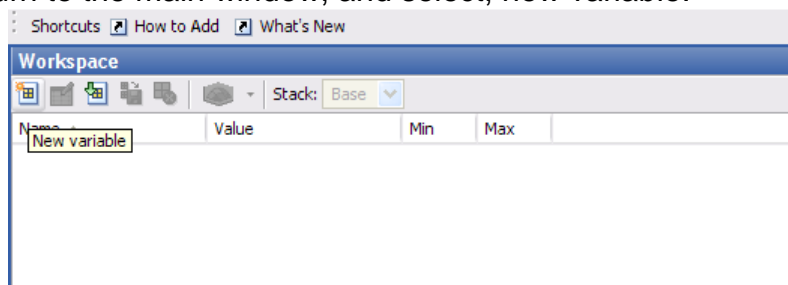
**Step 1:** Select the from file block in Simulink and place the block



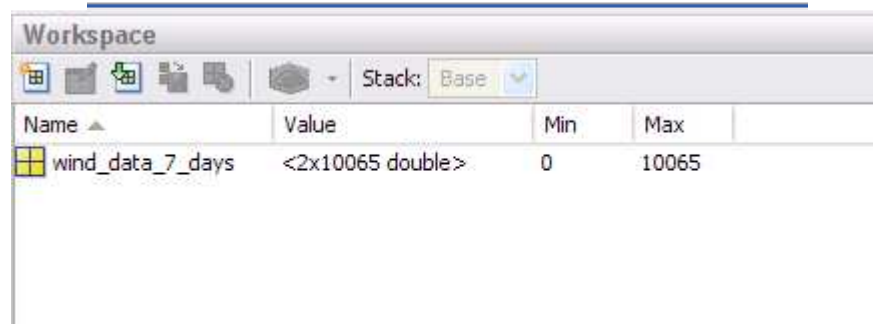
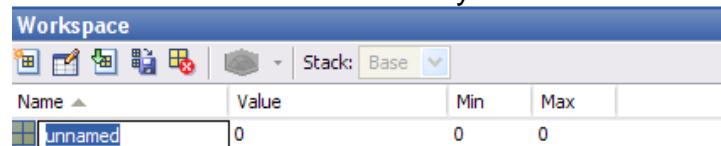
**Step 2:** Name the block the same name of the .mat file where the source data will be stored. Also, set the sample time of the data.



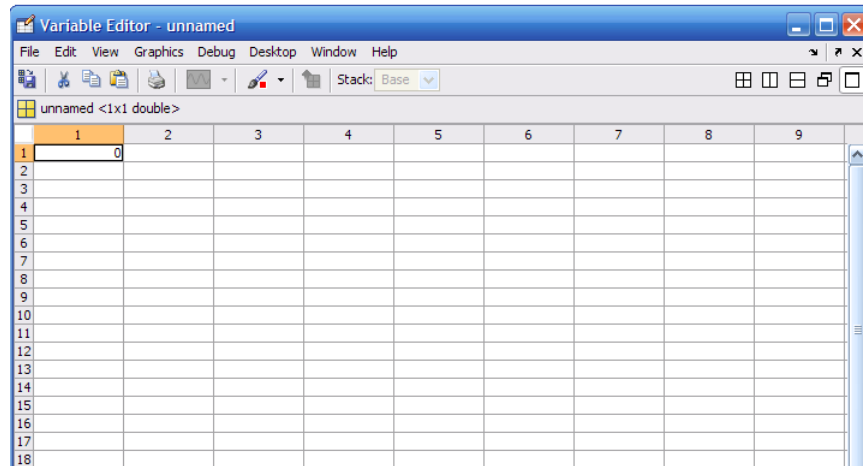
**Step 3:** Return to the main window, and select, new variable.



Name your variable the same name as the file you want to load from



**Step 4:** Now Double click on the variable and open it up, there should be nothing there.



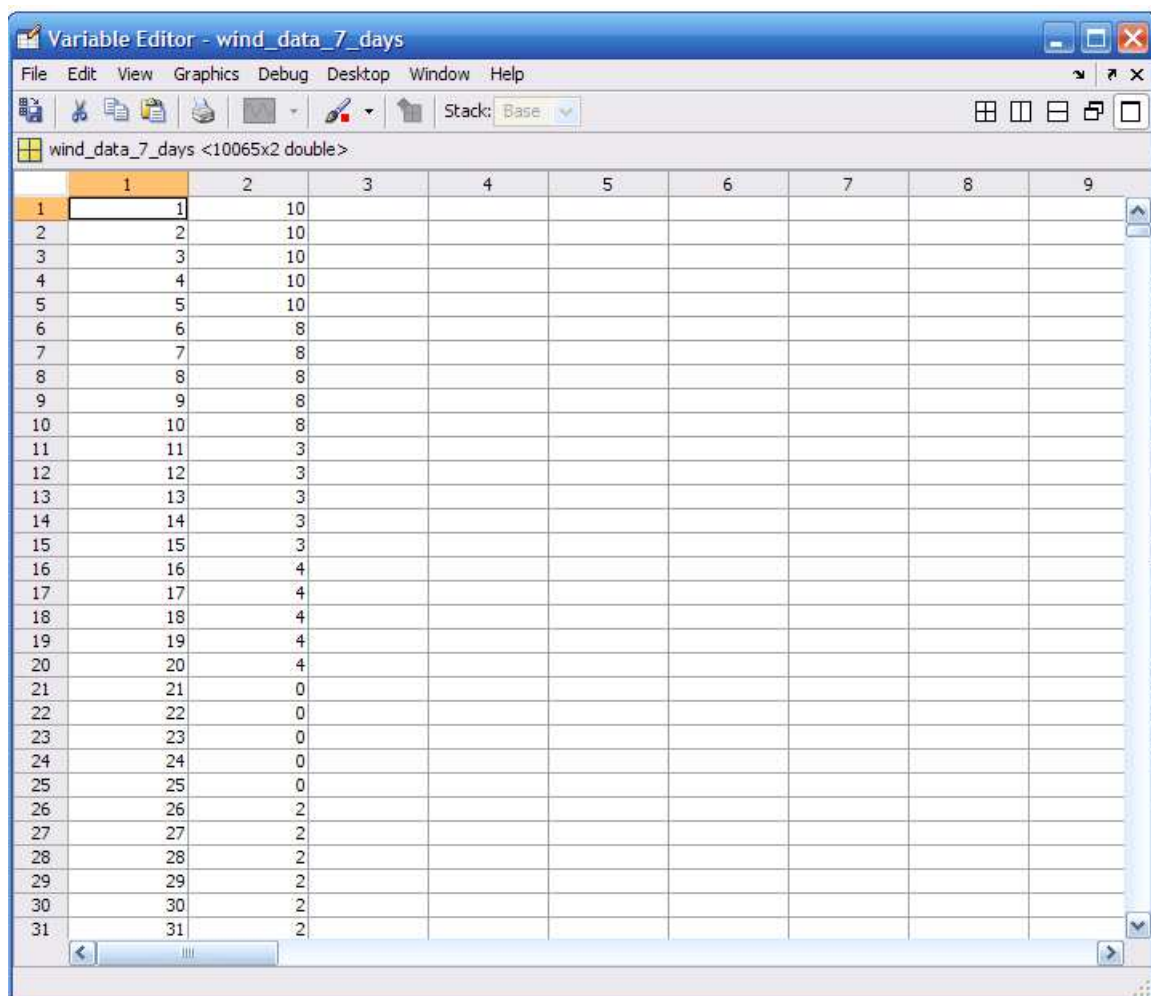
Variable Editor - unnamed

File Edit View Graphics Debug Desktop Window Help

unnamed <1x1 double>

	1	2	3	4	5	6	7	8	9
1	0								
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									

**Step 5:** Now you need to place your data into the variable. You must do this in a particular format.



Variable Editor - wind\_data\_7\_days

File Edit View Graphics Debug Desktop Window Help

wind\_data\_7\_days <10065x2 double>

	1	2	3	4	5	6	7	8	9
1	1	10							
2	2	10							
3	3	10							
4	4	10							
5	5	10							
6	6	8							
7	7	8							
8	8	8							
9	9	8							
10	10	8							
11	11	3							
12	12	3							
13	13	3							
14	14	3							
15	15	3							
16	16	4							
17	17	4							
18	18	4							
19	19	4							
20	20	4							
21	21	0							
22	22	0							
23	23	0							
24	24	0							
25	25	0							
26	26	2							
27	27	2							
28	28	2							
29	29	2							
30	30	2							
31	31	2							

Note: the left column always goes from 1 to max number of entries, this deals with the sample number. The right value is the data being sent.

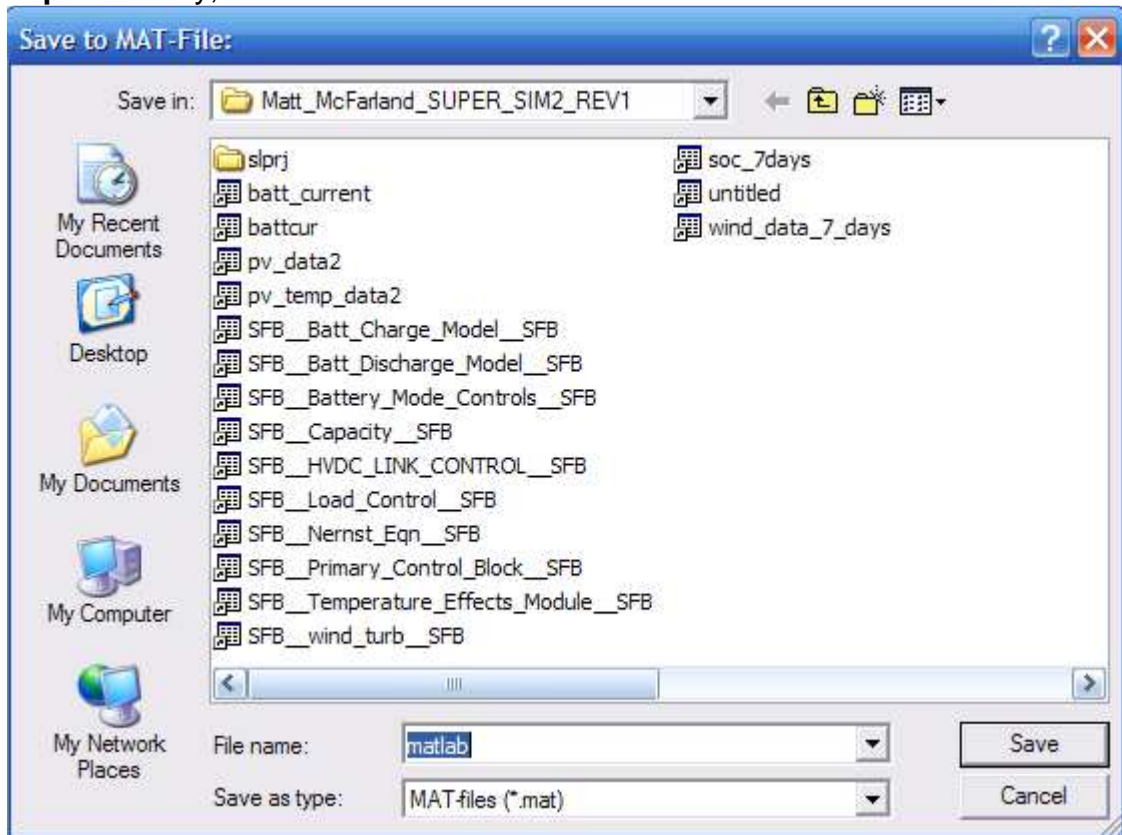
Save and close the data.

TYPE1	TV	Cooler	LED	DC Motor	Internet	TYPE2	TV	Cooler	LED	DC Motor	Internet
	0						0				
	100						100				
	200						200				
	300						300				
	400						400				
	500						500				
	600						600				
	700						700				
PV on	800	X	X		X	PV on	800				
	900	X	X		X		900				
	1000	X	X		X		1000			X	
	1100	X	X		X		1100			X	
	1200	X	X	X	X		1200			X	
	1300	X	X	X	X		1300			X	
	1400	X	X		X		1400				
	1500	X	X		X		1500				
	1600	X	X		X		1600				
	1700	X	X		X		1700				
PV off	1800			X		PV off	1800		X		
	1900			X			1900		X		
	2000			X			2000		X		
	2100			X			2100		X		
	2200						2200				
	2300						2300				

**Step 6:** Now in order for MATLAB to read the data correctly, the values must be transposed. This can be done simply by typing the following:

```
EDU>> wind_data_7_days=wind_data_7_days';
```

**Step 7:** Finally, select the variable and click save.

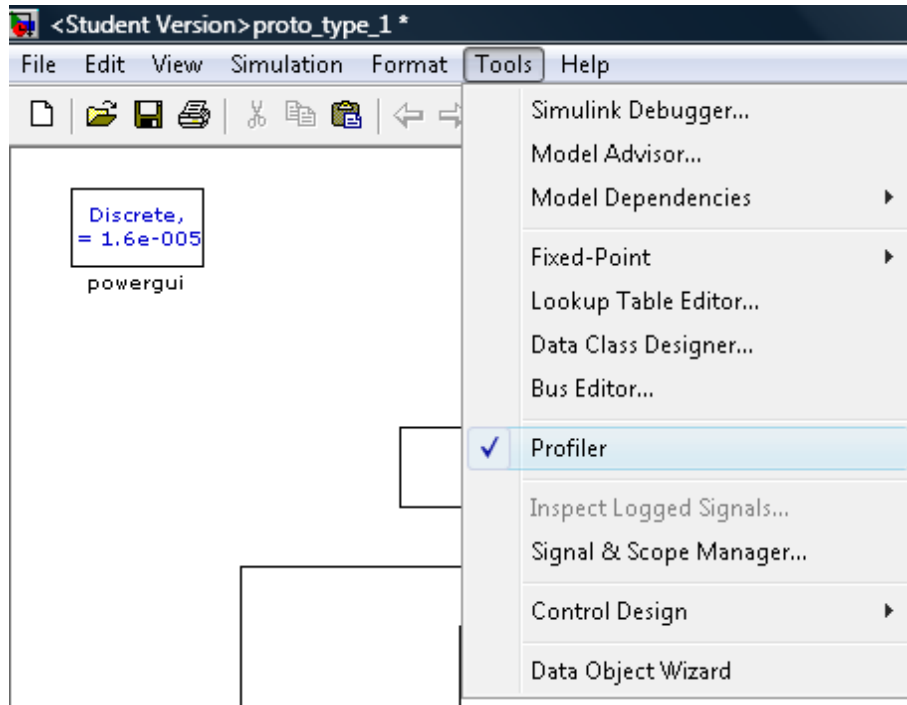


Save the variable the same name as the input file name you set in the beginning.

**Step 8:** Make sure the block is hooked up to something in Simulink and everything should work fine.

## Appendix D: Speed Testing

All Simulation times were measured using the Simulink profile report. To enable the profile report, do the following:



This will allow you to view a complete report after completion of running a simulation.

The profiler will allow you to see where your simulation may be slowing down and what blocks are taking up the most resources.

### D.1 DC-DC Converter Attempted New Model

#### Simulink Profile Report: Summary

*Report generated 22-Mar-2010 11:25:39*

Total recorded time:	798.55 s
Number of Block Methods:	108
Number of Internal Methods:	6
Number of Nonvirtual Subsystem Methods:	2
Clock precision:	0.00000005 s
Clock Speed:	2100 Mhz



As seen here it takes 798.55s or 13.3minutes of real time to simulate only 6 seconds of simulation time.

## D.2 DC-DC Converter Final Model Version

### Simulink Profile Report: Summary

*Report generated 21-Mar-2010 12:12:53*

Total recorded time:	7.14 s
Number of Block Methods:	153
Number of Internal Methods:	6
Number of Nonvirtual Subsystem Methods:	2
Clock precision:	0.00000005 s
Clock Speed:	2100 Mhz

As seen here it takes 7.14s of real time to simulate 600 seconds of simulation time.

In the first converter model, six seconds of simulation time was chosen because a longer simulation time would have taken a considerable amount of real time. Comparing the two converter models mathematically the ratio of real time to simulation time is 133 for the first DC-DC converter and .0119 for the final DC-DC converter. The final model is approximately 1000 times faster than the first case.

## Appendix E: Compiler Setup

- 1) In the MATLAB command prompt type: `mbuild -setup`
- 2) You should receive a prompt similar to the following:

Select a compiler:

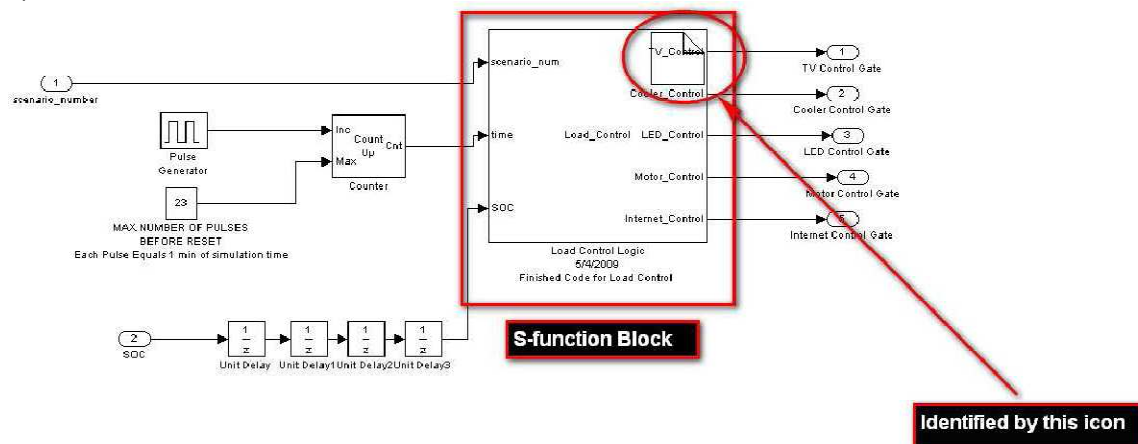
```
[1] Icc-win32 C 2.4.1
[2] Microsoft Visual C++ 6.0
[3] Microsoft Visual C++ 2005 SP1
[4] Microsoft Visual C++ 2008
[5] Microsoft Visual C++ 2008 Express
```

```
[0] None
```

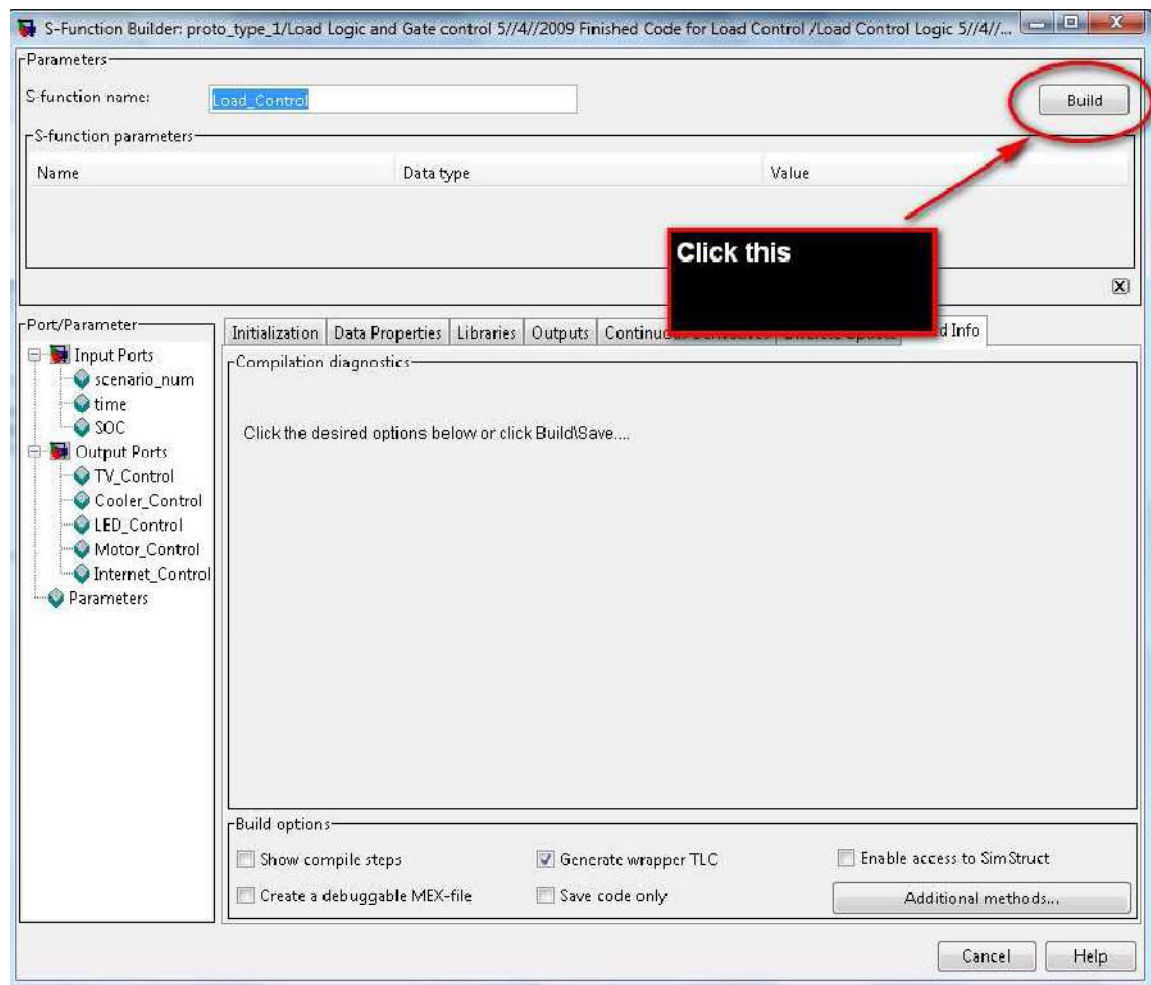
- 3) Select option 1 (this is usually the built in compiler for Mat lab)

*S-function REBUILD:*

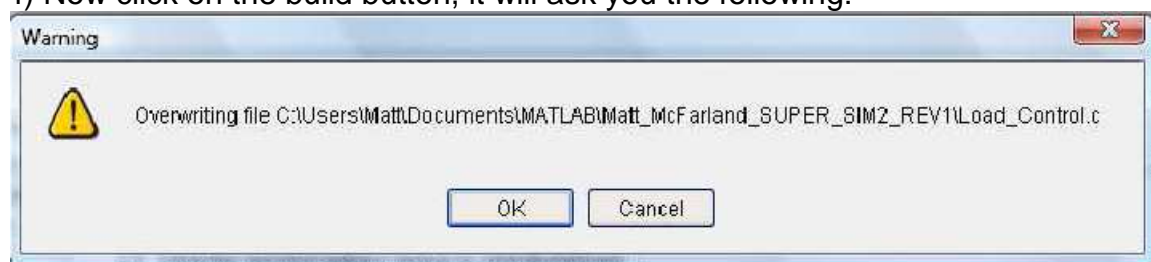
- 1) Find the S-function block:



- 2) Double click on the S-function Block
- 3) Now you should see the following:



4) Now click on the build button, it will ask you the following:



5) Click ok.

6) Next you will see the following:



7) Again click ok. You should see the following if everything worked right.

Compilation diagnostics

```
### 'Load_Control.c' created successfully  
### 'Load_Control_wrapper.c' created successfully  
### 'Load_Control.tlc' created successfully  
### S-function load_control.mexw32 created successfully
```

## **Appendix F: Readme File for DVD**

### CODE ON THE CONTROL LAPTOP:

- 1) ALL THE CURRENT CODE ON THE CONTROL LAPTOP AS OF 3/25/2010

### COMPONENTS:

- 1) BATTERY RESEARCH
  - a) Battery Academic Papers
  - b) Experimental Batt Data
  - c) Battery Datasheet
  - d) Battery Curve Characterization
- 2) WIND TURBINE
  - a) Turbine data sheets
  - b) Turbine academic papers
- 3) LOADS
  - a) Cell Phone
  - b) LED Lights
  - c) TE cooler Model
  - d) Spartan 3E
  - e) Internet
  - f) Radio
  - g) DC motor
  - h) Scenario Types
- 4) DC-DC Converter
  - a) All part datasheets
  - b) Efficiency data
- 5) Micro Grid
  - a) Microgrid Academic Papers
  - b) Voltage Selection

### INPUT DATA:

- 1) PV Data 7 days
  - a) PV data along with data conversion macro
- 2) Wind Data Cal Poly
  - a) RAW CAL POLY WIND DATA FILES
  - b) Processed Wind data files for the date range + VBA macros

### OUTPUT DATA:

- 1) Scenario 1
- 2) Scenario 2
- 3) Scenario 3
- 4) Scenario 4
- 5) Scenario 5
- 6) Scenario 6
- 7) Loading Graphs

### MATLAB MODELS:

- 1) OLD MODEL BY TYLER SHEFFIELD
- 2) NEW MODELS BY MATTHE MCFARLAND
  - a) Microgrid Model
  - b) Wind Turbine Model
  - c) Motor model
  - d) Battery Model
  - e) Thermo electrical cooler model
  - f) Full system model

### Demonstration Video for Running Simulation